ARC-VM: AN ARCHITECTURE REAL OPTIONS COMPLEXITY-BASED VALUATION METHODOLOGY FOR MILITARY SYSTEMS-OF-SYSTEMS ACQUISITIONS

A Thesis
Presented to
The Academic Faculty

by

Jean Charles Domerçant

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2011

ARC-VM: AN ARCHITECTURE REAL OPTIONS COMPLEXITY-BASED VALUATION METHODOLOGY FOR MILITARY SYSTEMS-OF-SYSTEMS ACQUISITIONS

Approved by:

Dimitri N. Mavris, Professor
Advisor and Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Vitali Volovoi, Assistant Professor
School of Aerospace Engineering
Georgia Institute of Technology

Santiago Balestrini-Robinson, Ph.D.
School of Aerospace Engineering
Georgia Institute of Technology

Brian J. German, Assistant Professor
School of Aerospace Engineering
Georgia Institute of Technology

Mrs. Kelly Cooper
Office of Naval Research

Date Approved: 14 November 2011

To my family, my friends, and to Doc.

ACKNOWLEDGEMENTS

The development of this thesis has been both rewarding and challenging. Without the aid of my family and friends, this effort would not have been as fulfilling. First and foremost, I would like to thank my mother, my brother, and my aunt for their steadfast dedication and support. Their belief in me was sometimes the only thing that sustained me through the rough patches. I would also like to thank all of my friends near and far for their continued encouragement and enthusiasm. This includes all of my friends and colleagues at ASDL, both past and present, with special mention to the ARCHITECT team. Special thanks to Joseph Iacobucci, Kelly Griendling, Burak Bagdatli, Annie Jones, and Daniel Cooksey for their valuable feedback, much needed criticism, and positive support. It has been both an honor and a pleasure being a part of the ARCHITECT team.

I would also like to express gratitude to my committee, Professors Vitali Volovoi and Brian German, Dr. Santiago Balestrini-Robinson, and Mrs. Kelly Cooper. Mrs. Cooper's sponsorship and perspective, in particular, proved invaluable. I also owe much gratitude to Dr. Balestrini-Robinson for his wise council over the many months it took for this thesis to take shape, and for providing an important sounding board to my ideas. Last, but by no means least, I would like to thank my committee chairman and advisor Dr. Dimitri Mavris. Over the past five years I have come to admire not only his vast engineering knowledge, but also his passion and dedication in developing scholars. Though he always kept it challenging, he also always managed to remind us that this process can be as fun as it is important. I will always be grateful for the chances you have given me, and thank you again for helping me to develop both professionally and personally.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAA Anti-Aircraft Artillery.

ACAT Acquisition Category.

AoA Analysis of Alternatives.

AOR Area of Responsibility.

AOS Acquisition Option Space.

ARC-VM Architecture Real Options Complexity-Based Valuation Methodology.

ARCNET Architecture Resource-Based Collaborative Network Evaluation Tool.

BMDS Ballistic Missile Defense System.

C2 Command and Control.

C4ISR Command, Control, Communications, Computing, Intelligence, Surveillance, and Reconnaissance.

CAIV Cost as an Independent Variable.

CBA Capabilities Based Assessment.

CBP Capability Based Planning.

CER Cost Estimating Relationship.

CJCS Chairman of the Joint Chiefs of Staff.

CNE Coefficient of Networked Effects.

CoBRA Complexity Based Risk Assessment.

CONOPs Concept of Operations.

COSYSMO Constructive Systems Engineering Cost Model.

CPL Characteristic Path Length.

DARPA Defense Advanced Research Projects Agency.

DCF Discounted Cash Flow.

DoD Department of Defense.

DOE Design of Experiments.

DOTMLPF Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel, and Facilities.

DP Design Parameter.

DSS Decision Support System(s).

EV Earned Value.

EVM Earned Value Management.

EWR Early Warning Radar.

FCS Future Combat Systems.

FDC Functional Distribution Complexity.

FPC Functional Processing Complexity.

FR Functional Requirement.

FV Fiedler Vector.

GAO Government Accountability Office.

GCI Ground Control Intercept.

IADS Integrated Air Defense System.

IOC Initial Operational Capability.

IOL Interoperability Level.

IPPD Integrated Product and Process Development.

IPT Integrated Product Teams.

IRL Integration Readiness Level.

IRR Internal Rate of Return.

IT Information Technology.

IW Irregular Warfare.

JCIDS Joint Capabilities Integration and Development System.

JOA Joint Operating Area.

LCC Life Cycle Cost.

LRS Long Range Strike.

MDA Milestone Decision Authority.

MoE Measure of Effectiveness.

MoP Measure of Performance.

M&S Modeling and Simulation.

NASA National Aeronautics and Space Administration.

NATO North Atlantic Treaty Organization.

NAV Net Acquisition Value.

NCW Network Centric Warfare.

NDS National Defense Strategy.

NMS National Military Strategy.

NPV Net Present Value.

NSS National Security Strategy.

OMB Office of Management and Budget.

O&S Operating or Operations & Support.

PAT Portfolio Analysis Tool.

PFE Perron-Frobenius Eigenvector.

PoPS Probability of Program Success.

PPBE Planning, Programming, Budgeting, and Execution.

PV Planned Value.

QDR Quadrennial Defense Review.

QFD Quality Functional Deployment.

RBN Random Boolean Network.

R&D Research and Development.

RDT&E Research, Development, Testing, and Evaluation.

ROA Real Options Analysis.

ROR Rate of Return.

RPC Resource Processing Complexity.

RSC Resource State Complexity.

RSM Response Surface Methodology.

RSS Resource State Specifier.

SAM Surface-to-Air Missile.

SEM Simplified Engagement Model.

SIAP Single Integrated Air Picture.

SoS System(s)-of-Systems.

SoSSE System-of-Systems Systems Engineering.

STANAG Standardization Agreement.

TBM-N Theater Ballistic Missile - Nuclear.

T&E Test and Evaluation.

TOC Total Ownership Cost.

TRL Technology Readiness Level.

UAV Unmanned Aerial Vehicle.

UNTL Universal Naval Task List.

U.S. United States.

WBS Work Breakdown Structure.

WWII World War II.

SUMMARY

The combination of today's national security environment and mandated acquisition policies makes it necessary for military systems to interoperate with each other to greater degrees. This growing interdependency results in complex Systems-of-Systems (SoS) that only continue to grow in complexity to meet evolving capability needs. Thus, timely and affordable acquisition becomes more difficult, especially in the face of mounting budgetary pressures. To counter this, architecting principles must be applied to SoS design.

The research objective is to develop an Architecture Real Options Complexity-Based Valuation Methodology (ARC-VM) suitable for acquisition-level decision making, where there is a stated desire for more informed tradeoffs between cost, schedule, and performance during the early phases of design. First, a framework is introduced to measure architecture complexity as it directly relates to military SoS. Development of the framework draws upon a diverse set of disciplines, including Complexity Science, software architecting, measurement theory, and utility theory. Next, a Real Options based valuation strategy is developed using techniques established for financial stock options that have recently been adapted for use in business and engineering decisions. The derived complexity measure provides architects with an objective measure of complexity that focuses on relevant complex system attributes. These attributes are related to the organization and distribution of SoS functionality and the sharing and processing of resources. The use of Real Options provides the necessary conceptual and visual framework to quantifiably and traceably combine measured architecture complexity, time-valued performance levels, as well as programmatic risks and uncertainties.

An example suppression of enemy air defenses (SEAD) capability demonstrates the development and usefulness of the resulting architecture complexity & Real Options based valuation methodology. Different portfolios of candidate system types are used to generate an array of architecture alternatives that are then evaluated using an engagement model. This performance data is combined with both measured architecture complexity and programmatic data to assign an acquisition value to each alternative. This proves useful when selecting alternatives most likely to meet current and future capability needs.

CHAPTER I

INTRODUCTION

Maintaining effective armed forces is an integral part of national security. This requires, among other things, arming both today's and future forces with the proper military weapons systems. These systems are crucial to achieving desired mission capabilities within a dynamic national security environment. Thus, investments in technologies, programs, and product support are necessary to develop state-of-the-art complex systems capable of aiding the Department of Defense (DoD) in achieving the National Security Strategy (NSS) of the United States (U.S.) [60].

Defense acquisition has a long history and major wars fought in the past had enormous impact on how defense acquisition has been conducted over the years. While World War I saw many improvements in the application of new technologies and processes such as aircraft, submarines, industrialism, and mass production [90], World War II (WWII) is often considered the first major turning point for the following reason [120]:

> Until World War II, weapons acquisition in the United States was more a political than a military problem. Shielded from large external threats, the country had no pressing need for sophisticated weapons; with few exceptions it was content to let European militaries take the lead in developing and fielding new weaponry.

WWII saw the advancement of many new and existing technologies, the most notable of which is the development of the atomic weapon. To win the war effort, the nation was forced to spend unprecedented amounts on national defense and to incur corresponding unprecedented deficits — a trend that continues to the present day [146]. The lasting mark on defense acquisition has been the continued emphasis on research and development (R&D) programs. This has also impacted the scientific community. This sustained research and development is pivotal in ensuring that rapid advancements in technology would be a necessary capability for world powers to dominate in a global context. After WWII, escalating tensions between the U.S. and the Soviet Union over the potential spread of communism created a tense national security environment. The year 1947 is generally regarded as the commencement of the Cold War with the Soviet Union. During this time the U.S. saw the emergence of a persistent, international security threat. Thus, weapons systems were developed almost exclusively against a Soviet threat counterpart [146]. In terms of defense acquisition, the previous wars emphasized “simplicity, reliability, and producibility” of weapons systems [146]. During the Cold War, however, state-of-the-art technological advances were rapidly applied as military interests expanded into new realms such as communications, spaceflight, microelectronics, astrophysics and a host of other fields [146].

Defense acquisition during this time catered to a threat-based, stove piped military decision-making environment where duplication of capabilities among the nation’s armed forces was commonplace [93]. Joint war fighting was not emphasized and each service independently assessed their Concept of Operations (CONOPs), warfighter needs, and capabilities. After the Cold War, however, the nature of the international security environment began to fundamentally change. The security environment is now extremely fluid, with continually changing coalitions, alliances, and partnerships. Also, the rise of transnational terrorism means that Irregular Warfare (IW) has emerged as a major and pervasive form of warfare. IW is typically characterized by a less powerful adversary seeking to disrupt or negate the military capabilities and advantages of a more powerful, conventionally armed military force.

Those who wage IW pose an asymmetrical threat that rely on guerrilla tactics and unconventional methods, including the use of nuclear, biological, and chemical weapons [94]. This has prompted a shift away from a threat-based approach to one that focuses more on the required capabilities necessary to neutralize such an asymmetrical threat. The emergence of technologies such as the Internet, GPS, wireless networking, and fiber optics accompanied this shift, resulting in systems that have become more interdependent and interconnected. This increased level of system coupling has been deemed necessary in order to achieve greater levels of functionality and performance [36]. For example, Information Technology (IT) has become an essential feature of today's governmental, civil, and commercial organizations and many of their associated products and processes. Improvements in both computer hardware and software, and the development of the World Wide Web have fundamentally changed how we store, retrieve, process, and disseminate data, information, and other resources on a global scale.

Warfare has also been significantly impacted. The ability to gain information superiority through the use of IT has radically shifted how command & control (C2) is achieved on the battlefield. The post-Cold War military of today also relies more heavily on the use of IT to realize enhanced battle space dominance [12, 44]. This is embodied in the concepts of Command, Control, Communications, Computing, Intelligence, Surveillance, and Reconnaissance (C4ISR) systems and Network Centric Warfare (NCW). NCW advocates the intelligent use of information sharing for increased situational awareness and the promotion of synergistic effects [12, 44, 68]. Add to this the nature of the post-Cold War security environment previously discussed, and the response by military planners has been to make joint operations between different branches of the armed forces the dominant military doctrine [46, 97]. Joint operations requires that military systems be able to interoperate with one another. Facilitating this interoperation creates a greater dependence on IT for communication and coordination [68]. The immediate result is that now, complex Systems-of-Systems (SoS) must be considered in the context of design to deliver capabilities against varying and adaptive threats.

The DoD defines a SoS as a "set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities" [1]. Alternately, a SoS can be considered as "Groups of systems, each of which individually provides its own mission capability, that can be operated collectively to achieve an independent, and usually larger, common mission capability" [10]. SoS are commonly associated with complex behavior. The following list is a compilation of system traits from various sources that commonly lead to complex behavior [33, 34, 59, 114]. These traits are directly applicable to military SoS:

- Operational & Managerial Independence of Elements
- Evolutionary Development
- Geographical Distribution of Elements
- Networks of Systems
- Inter-disciplinary Study
- Heterogeneity of Systems

Two examples of the types of complex SoS recently under development are the Army's Future Combat System (FCS), which is intended to be a modernization program consisting of a family of manned and unmanned systems connected by a common network, and the nation's Ballistic Missile Defense System (BMDS). A graphical overview of each is provided in Figures 1 & 2.

While SoS bring added capabilities to help achieve greater levels of military effectiveness, they also introduce additional complexities in the fielding and maintaining

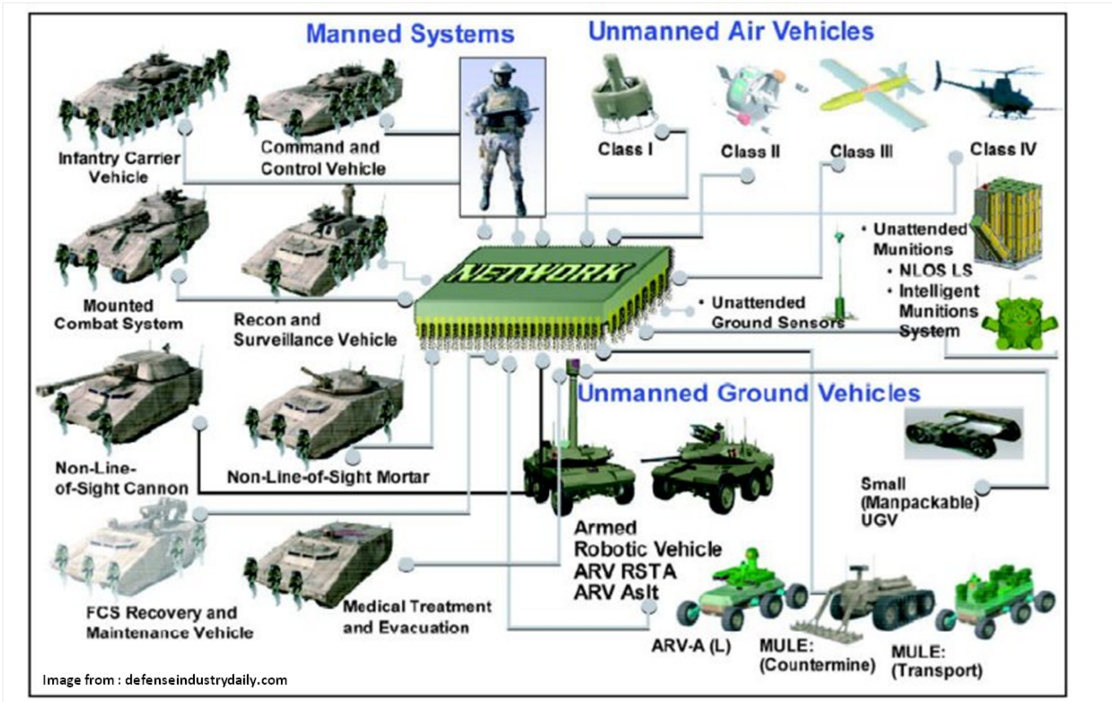


Figure 1: Example SoS: Army Future Combat Systems.

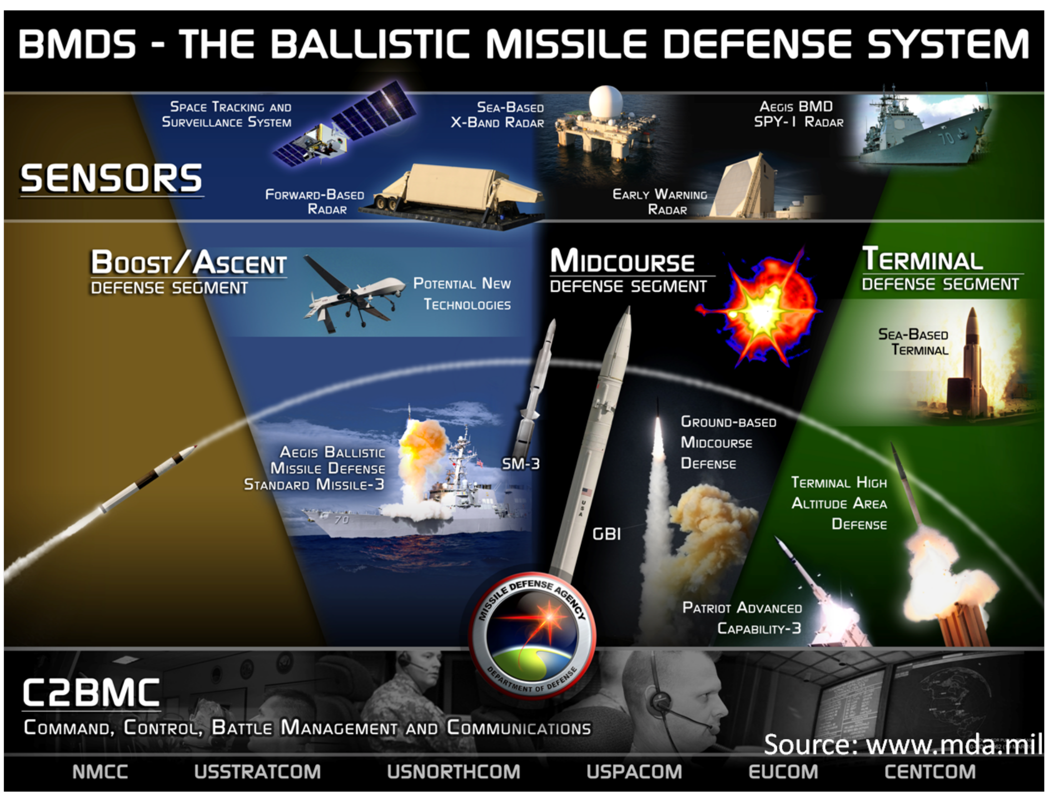


Figure 2: Example SoS: Ballistic Missile Defense System.

new weapons systems. In view of this, perhaps the greatest impact of the Information Age on engineering design is the heightened importance of understanding and managing the complexity that results from this new level of interconnectivity and interdependency [13]. The reason for this is that functional interoperation between systems requires complex information and resource exchanges, system interfaces, coordination, and added training to deliver full capabilities to the warfighter [64]. There is added difficulty when one considers that older so-called legacy systems must be incorporated into the mix as well. Legacy systems may possess outdated interfaces or technology, and different platforms of the same type may cover a spectrum of different upgrade versions. This means that functional interoperation must occur even when some assets may not have been designed for such from the outset. This also highlights the fact that different systems within the SoS will most likely be in different stages of their respective life cycles. Finally, there will be multiple stakeholders, oftentimes with conflicting interests. When all is said and done, the move towards SoS adds yet another layer of complexity to the acquisition problem. Figure 3 provides an excellent summary of this challenging problem domain.

Military planners quickly realized that a method to identify warfighter needs that cuts across all services would be more beneficial to fulfilling the NSS. This prompted a shift to a more top-down, capabilities-based approach to identify current gaps in capabilities across the joint war fighting areas. The Capabilities Based Planning (CBP) process encompasses the principal processes for transforming the nation's military forces to support the NSS. Once capability gaps have been identified and a materiel [1] solution deemed necessary, it is the role of defense acquisition to bring the needed systems and capabilities to the warfighter in a cost-effective, timely manner. However, capitalizing on and managing the complexities of CBP and NCW requires the

[1] *Materiel* is a generic word for equipment. It is inherently plural. It is distinguished from material, which is what things are made of [1].

Aspect of Environment	System	Acknowledged System of Systems
Management & Oversight		
Stakeholder Involvement	Clearer set of stakeholders	Stakeholders at both system level and SoS levels (including the system owners), with competing interests and priorities; in some cases, the system stakeholder has no vested interest in the SoS; all stakeholders may not be recognized
Governance	Aligned PM and funding	Added levels of complexity due to management and funding for both the SoS and individual systems; SoS does not have authority over all the systems
Operational Environment		
Operational Focus	Designed and developed to meet operational objectives	Called upon to meet a set of operational objectives using systems whose objectives may or may not align with the SoS objectives
Implementation		
Acquisition	Aligned to ACAT Milestones, documented requirements, SE with a Systems Engineering Plan (SEP)	Added complexity due to multiple system lifecycles across acquisition programs, involving legacy systems, systems under development, new developments, and technology insertion; Typically have stated capability objectives upfront which may need to be translated into formal requirements
Test & Evaluation	Test and evaluation of the system is generally possible	Testing is more challenging due to the difficulty of synchronizing across multiple systems' life cycles; given the complexity of all the moving parts and potential for unintended consequences
Engineering & Design Considerations		
Boundaries and Interfaces	Focuses on boundaries and interfaces for the single system	Focus on identifying the systems that contribute to the SoS objectives and enabling the flow of data, control and functionality across the SoS while balancing needs of the systems
Performance & Behavior	Performance of the system to meet specified objectives	Performance across the SoS that satisfies SoS user capability needs while balancing needs of the systems

Figure 3: Comparing the Acquisition of Systems vs. Systems-of-Systems [69].

further development of Systems Engineering (SE) practices beyond traditional SE approaches. The response has been the development of System-of-Systems SE (SoSSE) and the mandated use of architectural frameworks such as the Department of Defense Architectural Framework (DoDAF) [64, 69]. The SoS design framework aids in conceptualizing the organization and arrangement of systems that results when independent and useful systems are integrated into a larger system. This larger system is then able to deliver unique capabilities greater than the sum of the constituent parts. In conjunction, emerging SoSSE principles help to effectively manage the planning, analyzing, organizing, and integration of these emergent capabilities [69].

Since the threat environment is a dynamic one, existing and developing systems must constantly evolve to continue to provide value to the warfighter. SoS evolution can be properly managed through the use of architecting principles to adequately organize, describe, and maintain a complex grouping of systems. Within this context, the architecture defines the way in which the contributing, constituent systems work together. ***An architecture can be defined as a shared, persistent technical framework that governs the structure of components within the SoS, their relationships and dependencies, and the principles and guidelines governing their design evolution over time. An architecture includes not only systems and their functions, but data flow and communications protocols, key SoS functions, as well as end-to-end functionality. An architecture is used to address possible changes in functionality, performance, or interfaces*** [54, 64, 69]. The development of SoS architectures to manage the increased complexity of weapons systems underlines the growing challenges the acquisition community faces moving forward in acquiring new capabilities, especially in the face of mounting economic and budgetary pressures.

The research objective is to develop an Architecture Real Options Complexity-Based Valuation Methodology (ARC-VM) suitable for acquisition-level decision making, where there is a stated desire for more informed tradeoffs between cost, schedule, and performance during the early phases of design. It has been observed that decisions regarding the SoS architecture made early in the design process greatly influence the success of acquiring timely, affordable military SoS. First, a framework is introduced to measure architecture complexity as it directly relates to military SoS. Development of the framework draws upon a diverse set of disciplines, including Complexity Science, software architecting, measurement theory, and utility theory. Next, a Real Options based valuation strategy is developed using techniques established for financial stock options that have recently been adapted for use in business and engineering decisions. The derived complexity measure provides architects with an objective measure of complexity that focuses on relevant complex system attributes. These attributes are related to the organization and distribution of SoS functionality and the sharing and processing of resources. These attributes describe both the organization and allocation of the functionality required to meet stated capability needs as well as the allocation, movement, and processing of vital resources such as information. This gives architects the ability to make more informed tradeoffs when determining the number, types, and relative complexity of constituent systems under consideration, the degree of overlapping/redundant functionality present, and also the patterns and levels of collaboration that provide the greatest benefits. The use of Real Options provides the necessary conceptual and visual framework to quantifiably and traceably combine measured architecture complexity, time-valued performance levels, as well as programmatic risks and uncertainties.

An example suppression of enemy air defenses (SEAD) capability demonstrates the development and utility of the resulting architecture complexity & Real Options based valuation methodology. Starting with the stated capability need, a notional SEAD mission profile is decomposed into hierarchical activities and tasks. Then different portfolios of candidate system types are used to generate an array of architecture alternatives that can be evaluated using an engagement model. This performance data is combined with both measured architecture complexity and programmatic data to assign an acquisition value to each alternative. This proves useful when selecting alternatives most likely to meet current and future capability needs.

CHAPTER II

BACKGROUND

2.1 Defense Acquisition Decision Support Systems

2.1.1 Defense Acquisition Overview

Acquisition, as it is defined by the DoD, encompasses the design, engineering, construction, test and evaluation (T&E), production, sustainment, operations support and disposal of defense systems [38]. Thus, when acquiring new weapons and related items such as military cargo trucks, IT systems, processes, services, and end products for national defense, the entire life cycle of the system from "cradle to grave" must be taken into account. Acquisition is comprised of three major Decision Support Systems (DSS) as depicted in Figure 4.

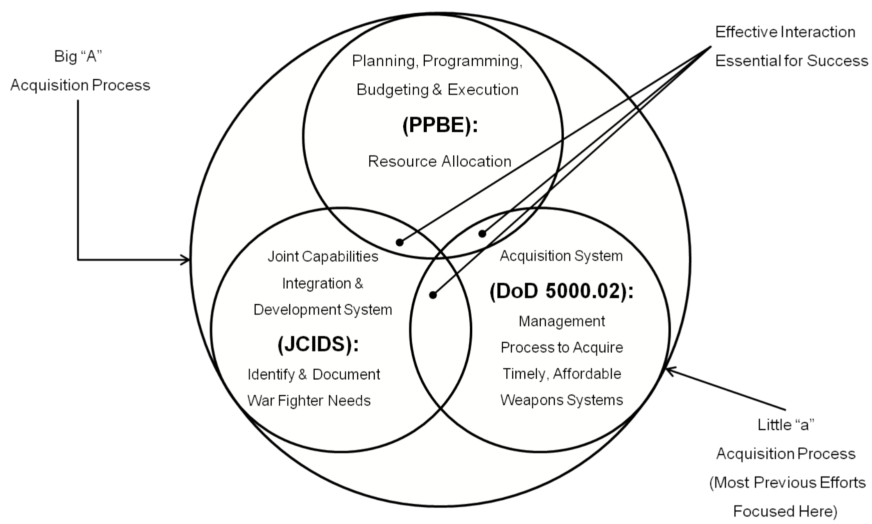


Figure 4: Defense Acquisition Decision Support Systems [16].

Commonly, a distinction is made between defense acquisition as a whole (often referred to as the 'Big A' encompassing all 3 DSS) and the Defense Acquisition System as a separate DSS [63]. As stated in DoD Directive 5000.01 [60]:

> The Defense Acquisition System exists to manage the nations investments in technologies, programs and product support to achieve the National Security Strategy and support the United States Armed Forces. The investment strategy of the Department of Defense shall be postured to support not only todays forces, but also the next force and future forces beyond that. The primary objective of defense acquisition is to acquire quality products that satisfy user needs with measurable improvements to mission capability and operational support, in a timely manner, and at a fair and reasonable price.

The Defense Acquisition System is the management process by which the DoD provides effective, affordable, and timely systems to the users [60]. Some of the policies that govern the Defense Acquisition System are:

- Flexibility
- Responsiveness
- Innovation
- Discipline
- Streamlined & Effective Management
- Cost & Affordability
- Cost Realism
- Integrated Test & Evaluation

DOD policy requires that a program manager be designated for each acquisition program in order to direct the development, production, and initial deployment of a new defense system. Program management represents the integration of a complex system of differing but related functional disciplines such as business and financial management, logistics, systems engineering, software management, T&E, manufacturing management, etc. These functional disciplines must work together to achieve program goals [60].

For management purposes, different acquisition categories (ACATs) are used to distinguish individual defense acquisition programs. Figure 5 provides a listing of the different categories as well as the criteria for designation and respective decision authority.

Category	Criteria for Designation	Decision Authority
ACAT I	• Major Defense Acquisition Programs - RDT&E total expenditure of more that $365M, or - Procurement total expenditure of more than $2.190B • MDA designation as special interest	• ACAT ID: USD(AT&L) - Reviewed by the Defense Acquisition Board (DAB) • ACAT IC: Component head, or Component Acquisition Executive (CAE) (cannot be further delegated) - Reviewed by component HQ
ACAT II	• Does not meet ACAT I criteria • Major System - RDT&E total expenditure of more than $140M, or - Procurement total expenditure of more than $660M • MDA designation	• CAE or the individual designated by the CAE • Reviewed IAW component policy
ACAT III Source: Defense Acquisition University	• Does not meet ACAT II or above criteria	• Designated by the CAE at the lowest appropriate level • Reviewed in accordance with component policy

Figure 5: Acquisition Categories for Weapons Systems (FY2000 Dollars) [60].

The different ACAT levels, which determine different levels of oversight and management processes, are principally based on dollar value level and level of milestone decision authority (MDA). When determining the level of oversight required, the risks associated with each program must be taken into account as well, though there is some criticism that the current ACAT paradigm does not go far enough in addressing these risks [125].

2.1.2 JCIDS: Joint Capabilities Integration Development System

The Joint Capabilities Integration Development System (JCIDS), created in 2003 to support the shift to a top-down, capabilities based approach, is the system responsible for identifying and documenting warfighter needs. Potential solutions to capability shortfalls can encompass any combination of changes to doctrine, organization, training, materiel, leadership & education, personnel, and facilities (DOTMLPF). Therefore, JCIDS is used to determine the best approach, materiel or otherwise, to meet existing capability gaps [46]. JCIDS is one component of the CBP process. Figure 6 shows the relationship between NSS, Capability Based Assessment (CBA), and JCIDS.

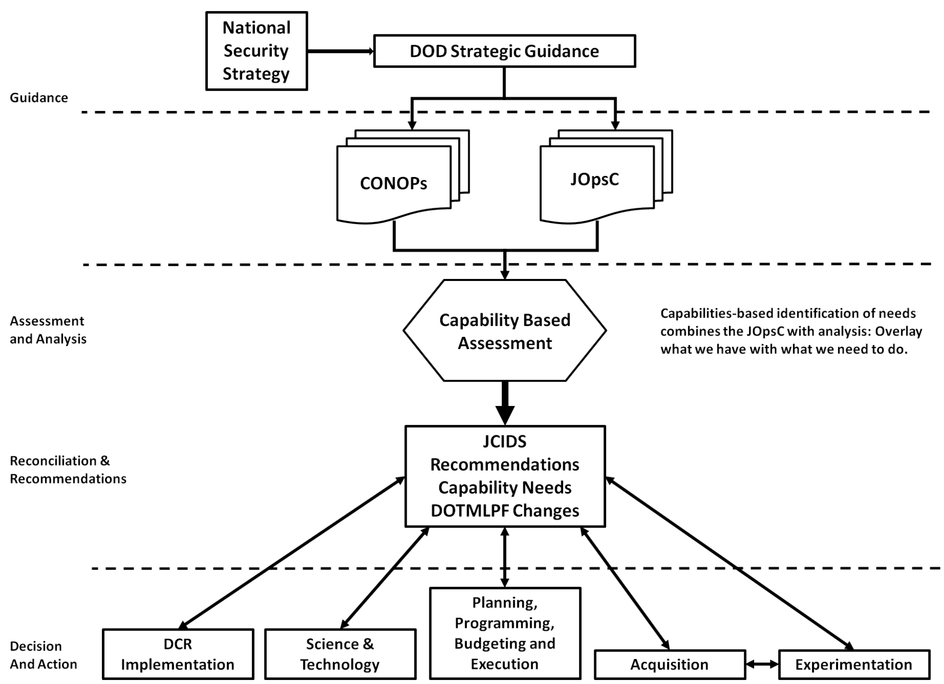


Figure 6: Top-Down Capability Needs Identification Process [45, 46].

The CBA forms the backbone of the JCIDS process and identifies required capabilities, shortfalls, operational risks due to lack of capabilities, and possible non-materiel approaches for eliminating shortfalls [60]. Once a materiel solution is deemed necessary, the Acquisition system is engaged to develop the appropriate system. Otherwise, changes to doctrine, organization, training, leadership & education, personnel, and/or facilities are requested.

2.1.3 PPBE: Planning, Programming, Budgeting & Execution

The Planning, Programming, Budgeting & Execution (PPBE) Process is the first phase in the allocation of resources to different programs responsible for developing needed weapons systems [38]. The 4 phases are:

- Phase 1-PPBE
- Phase 2-Enactment
- Phase 3-Apportionment
- Phase 4-Execution

PPBE is intended to provide the best mix of forces, equipment, and support within fiscal constraints and develops the proposed budget to the President for all acquisitions [146]. Figure 7 provides an overview of the resource allocation process.

PPBE is a calendar driven system that works in annual cycles. It is based on a series of national strategy documents, such as the President's NSS, the Secretary of Defense's National Defense Strategy (NDS), and the Chairman of the Joint Chiefs of Staff (CJCS) National Military Strategy (NMS). Every four years a review is conducted in the following manner [38]:

> The NSS and NDS provide the strategic framework for a congressionally directed Quadrennial Defense Review (QDR) that takes place every four years during the first year of a new presidential administration. The QDR report is provided to congress concurrent with the President's Budget in the 2nd year of a new administration. The PPBE process includes planning, programmatic and budgetary actions to implement the military force

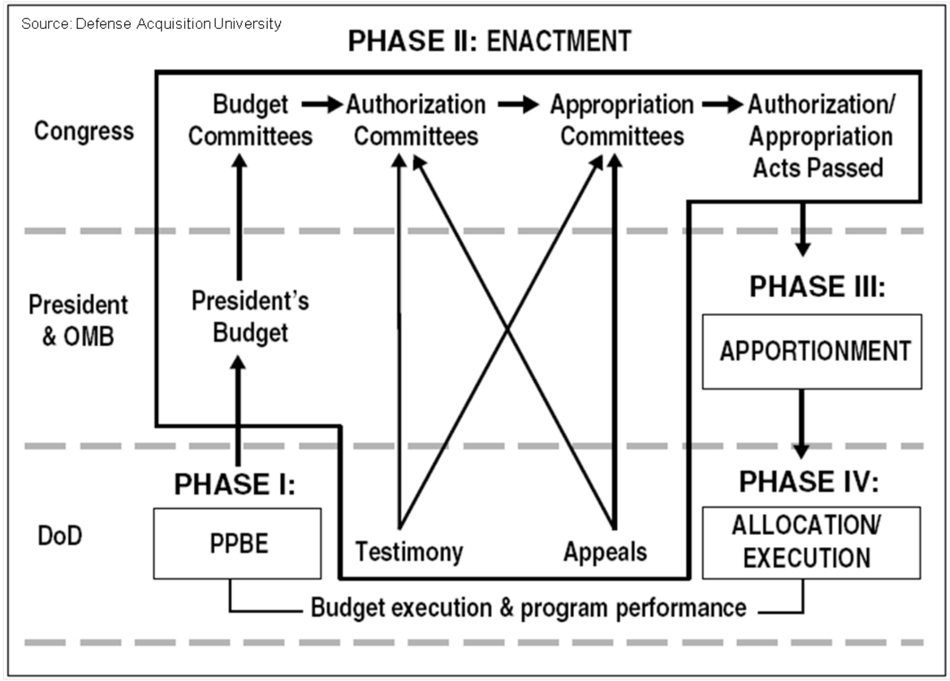


Figure 7: Resource Allocation Process.

structure and defense priorities outlined in the QDR report. The QDR occurs every four years. The NSS and NDS may be updated annually. The NMS is updated by the CJCS as necessary.

Once the President submits an annual budget for Congressional review the Enactment phase begins. Programs that receive approval and authorization also receive maximum funding levels and specifications on the quantities of systems to be procured. Next, the appropriations process provides budgetary authority to obligate and expend funds. During the apportionment phase the DoD and other federal agencies receive funds for further allocation within their respective departments. Finally, the appropriated funds are spent on defense programs during the execution phase. While planning is essentially a continuous process, many other actions run concurrently, for *e.g.,* the current fiscal year budget is being executed while enactment of next year's budget is under way, and programming for the following budget is in process [38].

2.1.4 Acquisition Life Cycle

The life cycle of a system from pre-system acquisition to operations & support (O&S) is separated into different phases by decision points called *milestones*. Three such milestones exist before a system reaches initial operating capability (IOC). These milestones are designated as Milestones A, B, and C. They are shown in Figure 8.

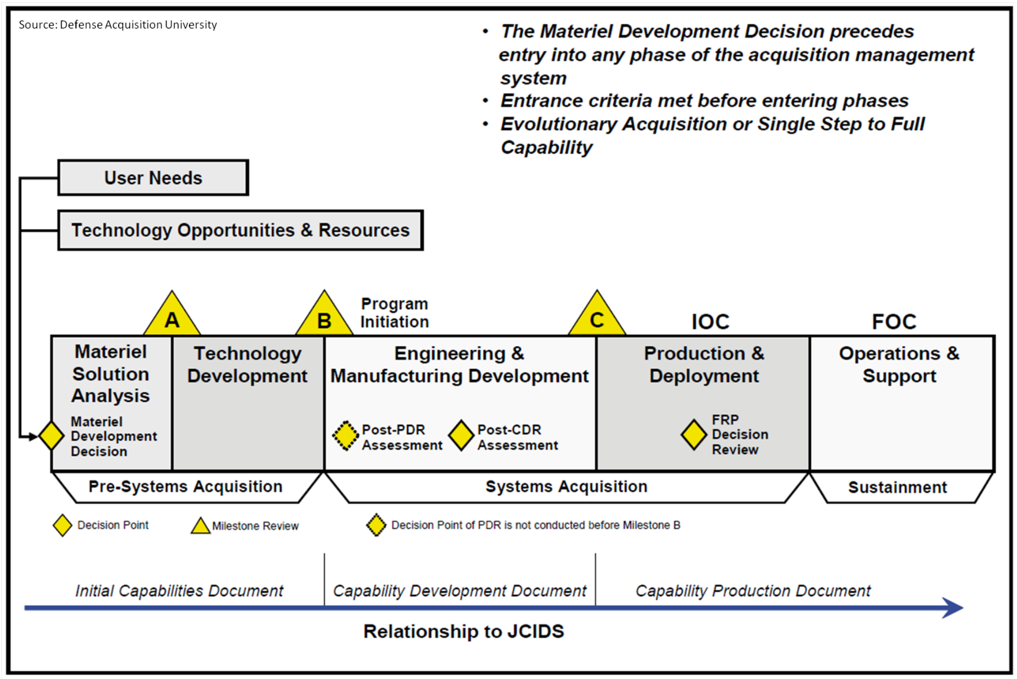


Figure 8: Acquisition Life Cycle.

The designated MDA is tasked with determining if the system satisfactorily meets certain entrance and exit criteria for continuation into the next phase of development. The following is a brief description of what occurs at each milestone [60]:

- Milestone A: The MDA approves a materiel development decision and grants formal entry into the acquisition process. This is a mandatory step for all programs. An Analysis of Alternatives (AoA) is conducted to determine which best provides the needed capability. Innovation and competition are emphasized and a technology development strategy is developed to help guide the efforts during the next phase. This phase ends when a recommendation is made for technological development of a materiel solution found using the AoA.

- Milestone B: This is normally the point of program initiation for defense acquisition programs. The MDA will confirm that the technology is mature enough for system-level development to begin. The MDA will also approve the acquisition strategy, acquisition program baseline, and types of contracts in place for the next phase of development before authorizing entry into the engineering and manufacturing development and demonstration phase. At this point the MDA must also certify that the program is affordable, has adequate funding, the technology has been demonstrated in a relevant environment, and that the acquisition program is likely to be successful.

- Milestone C: At this point, the MDA makes the decision to commit the DoD to production and authorizes entry into low-rate initial production or procurement. Transition from this phase means the system is ready to begin production and deployment and provide IOC, where a selected number of operational forces will receive the new system in order to conduct and support war fighting operations.

To support the program management decision-making process, credible and timely technical information that considers all of the life cycle needs must be furnished to decision makers at each Milestone. To accomplish this, the DoD acquisition process is critically dependent on effective and rigorous SE processes. SE is a standardized, disciplined management process that organizes and coordinates all engineering efforts toward the development of system solutions in an environment of change and uncertainty [65]. SE is applied at the initial stages of program formulation and continues throughout the system's life cycle to transform needed operational capabilities into an integrated system design. Without this, operationally affordable and sustainable weapon systems cannot be built [1].

2.2 Department of Defense Architecture Framework

2.2.1 DoDAF Development Timeline

As the acquisition environment for obtaining systems and capabilities becomes more dynamic and integrated, a method for managing the large amounts of data and information that describes the relationships between material, processes, and organizations internal and external to the DoD takes on even greater importance. One of the key means for ensuring interoperable and cost-effective military systems is to establish comprehensive architectural guidance for the entire DoD [62]. Therefore, the purpose of defining an architecture framework is to ensure the standardized development of architectures that promote interoperability across capabilities and among *integrated* architectures. The use of the word integrated in this context means that data required in more than one of the architectural models is commonly defined and understood across the models [64]. In this sense, a framework can be thought of as a computational or other modeling environment that allows analysis of an architecture, or put another way, the framework is used to view an architecture [31]. DoDAF serves as the principal guide for the development of integrated architectures and its use is mandated by law, policy, and guidance (Clinger-Cohen Act of 1996, Office of Management and Budget Circular A-130, and the 2004 DoDAF Promulgation Memo) [69]. Development of DoDAF began with the C4ISR Architecture Framework released in June 1996 in response to the Clinger-Cohen Act. Its primary purpose was to define and develop a better means and process for ensuring that C4ISR capabilities were interoperable and met the needs of the warfighter. Later, the C4ISR Framework was restructured into DODAF to broaden its applicability to all mission areas, not just the C4ISR community [62]. Figure 9 provides a timeline of DoDAF's development [62]. DoDAF V2.0 was recently introduced in May 2009.

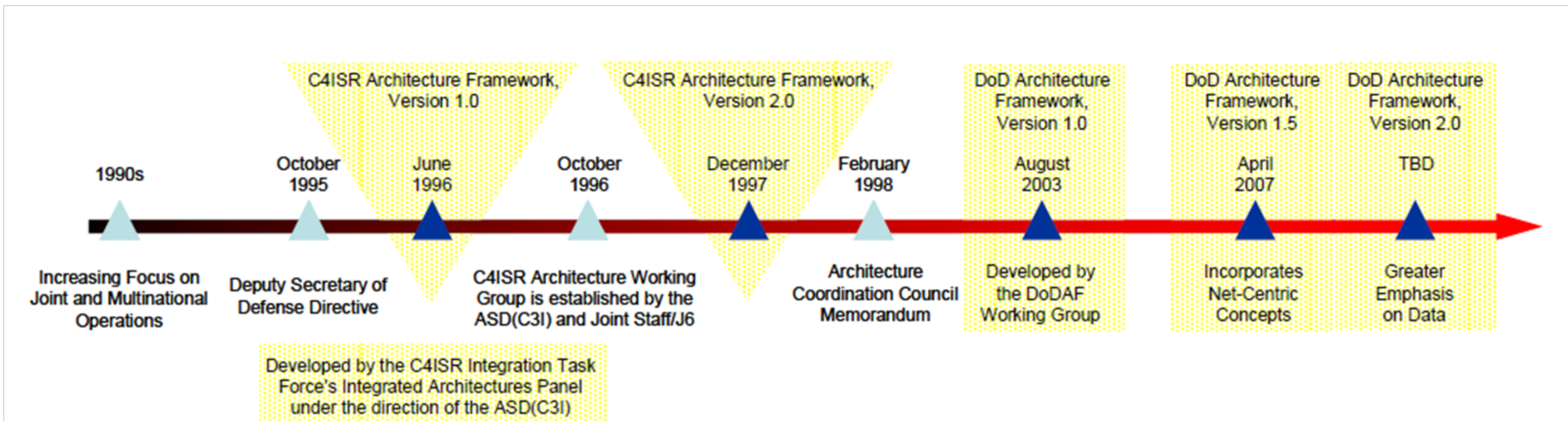


Figure 9: DoDAF Development Timeline [62].

2.2.2 DoDAF V2.0

In general, DoDAF describes different architectural views that help define the structure for organizing architecture concepts, principles, assumptions, and terminology about operations and technology into meaningful patterns to satisfy specific DoD purposes [64]. Prior to DoDAF V2.0 data was organized into "products" such as graphical representations or documents. DoDAF V2.0 places greater emphasis on a "data-centric" approach to organizing architectural data in supporting analysis and decision making. The primary motivation behind this approach is to provide renewed emphasis on Net-Centricity, or the ability to provide a framework for full human and technical interoperability to facilitate enhanced information sharing and protection.

Thus, DoDAF V2.0 is organized around data, models (templates for collecting data), and views (a representation of data in any understandable format) [64]. DoDAF V2.0 also places greater emphasis expanding the types of graphical representations that can be used to support decision-making. In comparison to DoDAF V1.5, DoDAF V2.0 specifies 2 new objects, the viewpoint and the viewpoint definition. A viewpoint describes data drawn from one or more perspectives and organized for useful decision making by management. Moreover, a viewpoint definition includes the information that should appear within individual views as well as how to construct and use the views. The viewpoint definition also includes the modeling techniques for expressing and analyzing the necessary information and a description of the purpose of the view and its intended audience. In all, over 50 different models exist for expressing the 8 different viewpoints. Figure 10 provides a description of the 8 different viewpoints and their relationship to DoDAF V1.5 products [130]. Figure 11 depicts the relationship among the different architecture methods, data, and presentation techniques.

An example of an operational view from the Operational Viewpoint is provided in Figure 12. This figure shows an OV-1, or a high-level operational concept graphic for providing a Single Integrated Air Picture capability (SIAP). This operational concept

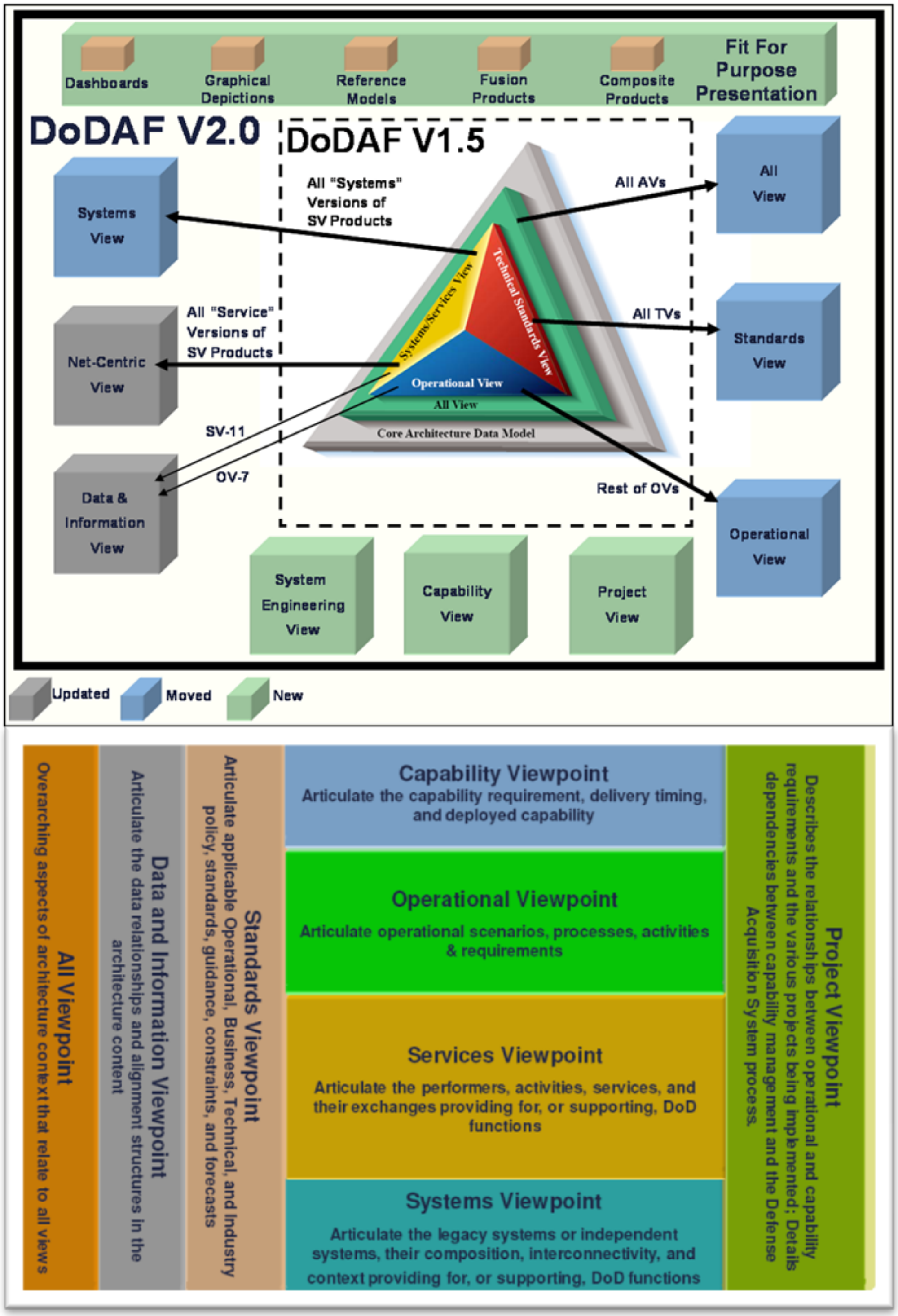


Figure 10: DoDAF Version 2.0 [64].

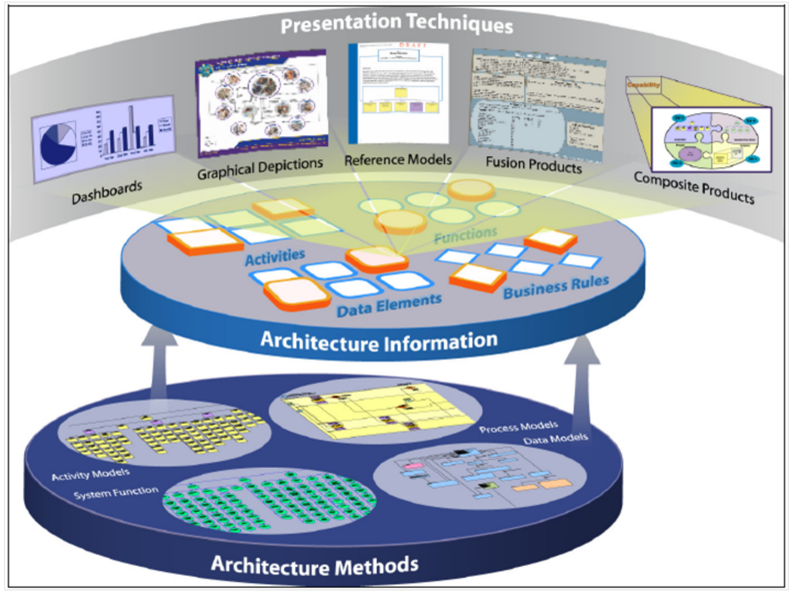


Figure 11: Relationships Between Architecture Methods, Data, and Presentation Techniques [64].

makes use of a combination of air, sea, land, and space-based military assets. Figure 13 is an example of an OV-2, or operational resource flow description. This view provides a description of the resource flows exchanged between operational activities. Many of the other views that can be created are capable of containing much more information than the views presented here, making them quite complicated. For example, there are various Standards, Systems, and Services views that seek to capture information such as technical standards, system-to-system interconnectivity & supporting automated systems, system functionality descriptions, and system resource flows. Depending upon the architecture being modeled, the construction of views using every single model from the 8 different viewpoints may not be necessary in describing the architecture, yet the sheer multitude of different models highlights the challenges in fully understanding and presenting the different interrelations, interactions, and interfaces a collective group of systems may posses within an architecture.

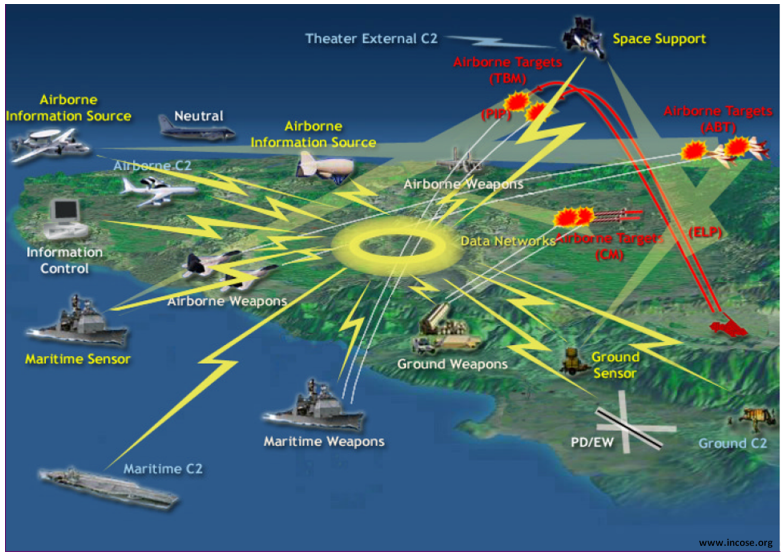


Figure 12: SIPA OV-1 High Level Operational Graphic.

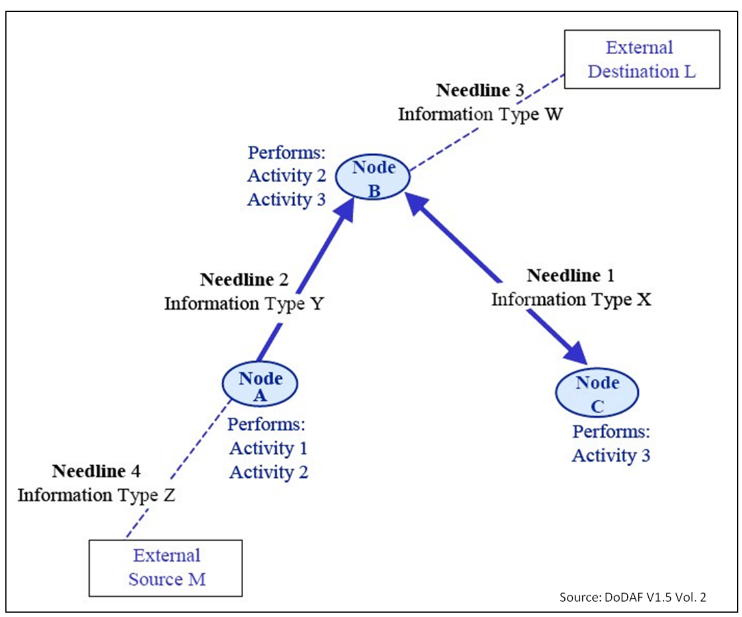


Figure 13: Example OV-2 Operational Resource Flow Description.

2.3 Ongoing Challenges to Successful Acquisitions

2.3.1 Cost, Schedule & Performance Tradeoffs

The success of an acquisition program is traditionally judged using cost, schedule, and performance as the main criteria [60]. In the past, when budgetary resources for defense spending were readily available, performance was the dominant criterion [81]. A 1986 study [141] found that "when acquisition problems arise, cost is the constraint most easily relaxed and schedule is next, whereas performance goals are adhered to most closely." Tyson *et al.* reported in 1992 that 66 out of 82 programs examined had experienced cost growth [156]. The current acquisition environment is also characterized by the frustration of the White House and Congress over cost overruns, schedule delays, and misdirected spending on major weapons programs. Recent studies by the Government Accountability Office (GAO) found that out of the 96 programs in the DoD's fiscal year 2008 portfolio of major defense acquisition programs, 42% of those programs had 25% or more increase in program acquisition unit cost by 2009 [86]. This is reflected in Figure 14. Moreover, the average delay in delivering initial capabilities has grown to 22 months for for the 2010 portfolio of major defense acquisition programs [87].

Analysis of DOD Major Defense Acquisition Program Portfolios (Fiscal Year 2009 Dollars)

Portfolio status	Fiscal year 2003 portfolio	Fiscal year 2007 portfolio	Fiscal year 2008 portfolio
Number of programs	77	95	96
Total planned commitments	$1.2 trillion	$1.6 trillion	$1.6 trillion
Commitments outstanding	$724 billion	$875 billion	$786 billion
Change to total research and development costs from first estimate	37 percent	40 percent	42 percent
Change in total acquisition cost from first estimate	19 percent	26 percent	25 percent
Estimated total acquisition cost growth	$183 billion	$301 billion[a]	$296 billion
Share of programs with 25 percent or more increase in program acquisition unit cost	41 percent	44 percent	42 percent
Average delay in delivering initial capabilities	18 months	21 months	22 months

Source: GAO analysis of DOD data.

Figure 14: GAO Analysis of DoD Cost & Schedule Overruns [86].

Perhaps this current frustration is amplified by the nostalgia of the past [10]:

> During World War II and the early Cold War, programs such as the Manhattan Project, the Defense Support Program (DSP), the intercontinental ballistic missile, and the U-2 surveillance aircraft all delivered very quickly, generally in fewer than 6 years, first products that today would be described as major systems. Currently, such major programs would likely require 10 to 20 years to complete.

Add to this that "The Apollo program in the 1960s, arguably one of the most complex space programs ever, took fewer than 8 years to complete" [10]. Tyson *et al.* provide a concise summary of the negative impacts of cost and schedule growth in the following statement [156]:

> Excessive schedules have two significant negative effects: U.S. forces may be left without needed capabilities and longer schedules often mean higher costs.

The authors go on to state that:

> Cost growth forces the DoD to revise budget plans, makes systems less affordable, and frequently erodes congressional support for acquisition programs.

In 2009 Defense Secretary Robert Gates was quoted as saying that the "spigot of defense spending that opened on 9/11 is closing" and he predicted "hard choices" [80]. Soon after, in May 2010, the Office of Management and Budget (OMB) provided a detailed summary of the billions of dollars in budget cuts recommended by Defense Secretary Gates. Among some of the programs that would be terminated or see significant program budget reductions were the C-17 air lifter, the F-22 Raptor fighter aircraft, Future Combat Systems, and Airborne Laser program, to name a few [43, 128].

2.3.2 Past Acquisition Reform Efforts

The DoD has a long history of acquisition reform efforts. For the most part, these reform efforts have been in response to governmental reports aimed at identifying and providing recommendations for seemingly chronic cost and schedule overruns [16, 146]. Past recommendations for reform all have in common their reiteration for far-reaching and significant changes to improve the overall process. Collectively, these reports have focused on nearly every aspect of defense acquisition from the size, culture, and training of the acquisition workforce to improving the contractual relationships between the DoD and defense industry suppliers, to restructuring the budgetary landscape. As a direct result the defense acquisition community has adopted a number of policies, practices, and project management techniques over the years aimed at helping it meet its overarching goal of delivering needed weapons systems on-time and within budget. One example is the shift towards design for affordability, where affordability is defined by the DoD as the degree to which the life cycle cost (LCC) of an acquisition program is in consonance with the DoD's long-range investment and force structure plans [1, 86, 117, 140].

Treating cost as an independent variable (CAIV) is another example of a relatively recent acquisition reform effort [1, 81]. The goal of CAIV is to reduce LCC especially in the early phases by performing cost/tradeoff analyses. Here, cost is held fixed, allowing performance and schedule to vary somewhat in an attempt to keep weapon systems affordable. As an acquisition process, CAIV is intended to integrate proven successful business-related practices with promising new DoD initiatives to obtain superior, yet reasonably priced war fighting capabilities [81]. This includes the mandated use of private sector management techniques such as Integrated Product and Process Development (IPPD) implemented by multifunctional, multidisciplinary Integrated Product Teams (IPT) that organize for and accomplish tasks that acquire goods and services [129]. The IPTs have at their disposal other quality management techniques pioneered in the private sector such as Quality Functional Deployment (QFD) for understanding user requirements and Six Sigma for managing quality and minimizing variability in manufacturing and business processes [37].

A last example is the use of Earned Value Management (EVM), a program management tool, whose implementation is even mandated for ACAT I programs [1]. EVM utilizes a work breakdown structure (WBS), or hierarchical product, data, and/or service decomposition of a project to track weekly progress. A valuation of planned work, termed planned value (PV), is assigned to each element in the WBS:

$$EV = \sum_{Start}^{Current} PV\,(Completed) \tag{1}$$

In this way, the earned value (EV) of completing certain project tasks can be tracked and measured. The primary benefit of this approach is that it acts as a diagnostic tool to alert program managers to any significant problems impeding progress. However, some of the major limitations and criticisms of EVM is that it is a measurement of project *quantity*, not quality, and that though preferred, it does not require precise, quantifiable measures for assessment of work completed [150]. This means that completion estimates may not accurately reflect the true state of the acquisition program, leading to decisions based on inaccurate assessments.

2.3.3 Weapons Systems Acquisition Reform Act of 2009

Part of the purpose for the development of the many tools, processes, and methods adopted and developed by the DoD is to aid acquisition analysts and decision makers in analyzing and selecting the best alternatives for future development. Based on historical trends, it remains to be seen the degree to which these more recent efforts will be successful in curbing the excessive cost and schedule overruns experienced in the acquisition of major weapons programs. Nonetheless, the task of delivering cost-effective weapons to the warfighter in a timely manner continues to take on even greater importance in the current national security environment. This has recently prompted Congress to pass legislation aimed at pushing for greater tradeoffs between cost, schedule, and performance at early stages in the design/acquisition process [80]. The result is the Weapons Systems Acquisition Reform Act of 2009 (S.454/P.L. 111-23) which became public law on May 22 of that year. One of the major provisions of the act is that it makes it more difficult for programs experiencing significant cost and schedule growth to pass key acquisition milestones. Another key provision is the appointment of a Director of Cost Assessment and Program Evaluation within the DoD. This person is responsible for for issuing policy and guidance on cost estimating and for developing confidence levels for those cost estimates. The bill also appoints a director to oversee developmental T&E as well as a director in charge of overseeing SE practices [146].

Undoubtedly, past recommendations such as providing a highly competent and stable workforce will add tremendously to achieving the goals of the acquisition community. Yet the recent reform act clearly illustrates that there is a strongly recognized need within the acquisition community for further improvements to current SE methods, especially when confronted with acquiring new weapon systems that must operate as part of one or more military SoS [5, 20]. Consequently, an initiative is underway within industry, academia, and government to refine guidelines and methodologies for systems engineers and architects. This is evidenced by the development of the DoD's 2008 *Systems Engineering Guide for Systems of Systems* [69]. This guide states that, "Most military systems today are part of an SoS even if they are not explicitly recognized as such. Operationally, the DoD acts as an SoS as the battle space commander brings together a mix of systems in an operation to meet mission objectives."

2.4 Analysis of Alternatives

2.4.1 AoA & Evolutionary Acquisition Strategy

The mounting pressures affecting today's acquisition environment further emphasizes the importance of successfully analyzing different SoS architectures under consideration. It is also important to consider that the challenges faced when designing/acquiring weapons systems are not strictly limited to the higher-level scope of SoS design problems. On the contrary, the challenge is compounded when one considers that these same design challenges affect a diverse array of individual systems ranging from automobiles to aircraft [50, 67, 83]. This is especially true when the design solution must be found outside of the historical database to provide new and evolving capabilities, and also relevant when these systems must now collectively work together within a unified architecture. Conklin eloquently describes this fundamental, polarizing nature of design [50]:

> Any design problem is a problem of resolving tension between what is needed and what can be done. On the one hand, the process of design is driven by some desire or need—someone wants or needs something new. The need might be expressed by a customer, or it may be a guess about what the market wants. The need or want is expressed in the language of what ought to be—what should be done, what should be built, what should be written.
>
> On the other hand, the process of design is constrained by resources—what can be done given the available resources such as time and money and the constraints imposed by the environment and the laws of science. Every need has a price tag—the process of design is about devising solutions that are feasible and cost effective.
>
> When an individual does design, she stands with one foot in each world.

> Moving back and forth between the two worlds, she tries to create a solution that joins the two polarities of design in an elegant way.

Thus, an AoA is a critically important element of defense acquisition. An AoA is formally defined as an analytical comparison of each alternative to determine its suitability in terms of the costs, operational effectiveness in meeting a capability need, and risks associated in designing, fielding, and maintaining that capability [1, 127]. Costs can be expressed as either life cycle or total ownership cost [1] (TOC). The AoA is part of the materiel solution analysis to identify the most promising end-state materiel solution for further technological development at Milestone A, as well as the best possible path to achieving that end-state solution. This is reflected in Figure 15 [1].

In order to reduce cycle time and speed the delivery of an advanced capability to warfighters, the DoD has identified evolutionary acquisition as the preferred strategy for achieving the end-state solution. Evolutionary acquisition is defined as a strategy that defines, develops, acquires, and fields an initial hardware or software increment (or block) of operational capability. Using this approach, a needed operational capability is met over time by delivering the capability in several increments. This means that future capability requirements are needed as the first increment may only deliver 60% to 80% of the desired final capability. But it also allows for the rapid insertion of new technologies as they develop. Overall, this approach requires collaboration among the user, tester, and developer to be successful [60, 74]. There are a number of DoDAF models designed to support the implementation of an evolutionary acquisition strategy. One example is the CV-3, or Capability Phasing model. The CV-3 tracks the development of a capability over time. Another is the SV-8, or Systems Evolution Description, which portrays the planned incremental steps toward migrating a suite of

[1] The TOC includes the elements of a program's LCC, as well as other related infrastructure or business process costs not necessarily attributed to the program in the context of the defense acquisition system [1].

systems to a more efficient suite, or toward evolving a current system to a future implementation. A final example is the SV-9, or Systems Technology & Skills Forecast. The SV-9 keeps track of the emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future system development [64].

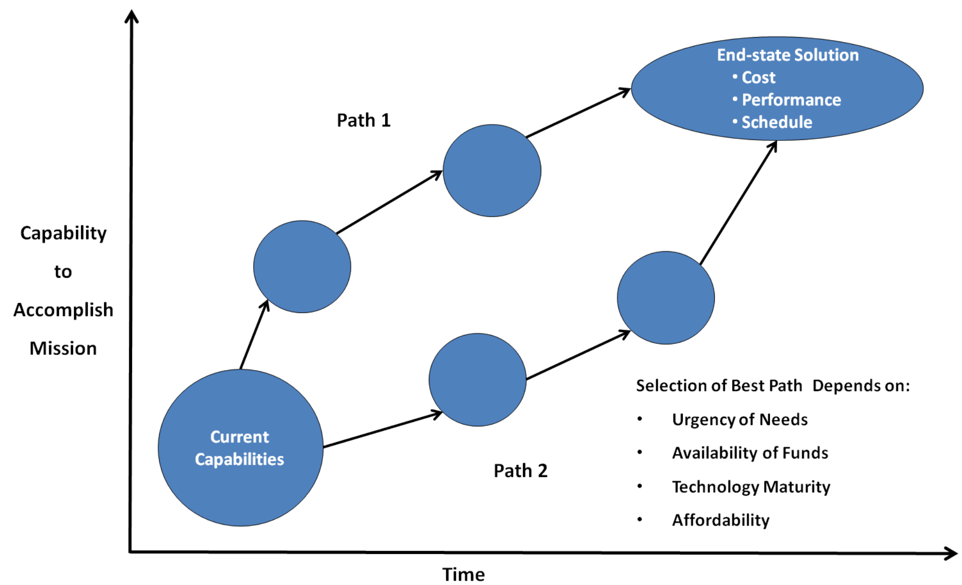


Figure 15: Establishment of an Evolutionary Acquisition Strategy. Image adapted from [1].

Developing a cost-effective, evolutionary strategy also depends upon a balanced assessment of a number of factors which should be addressed during the AoA [127]:

- What alternatives provide validated capabilities?
- Are the alternatives operationally effective and suitable?
- Can the alternatives be supported?
- What are the risks (technical, operational, programmatic) for each alternative?
- What are the life cycle costs for each alternative?
- How do the alternatives compare to one another?

In addition to the items mentioned above, scheduling is also a key consideration along with the scenarios and threats used during the analysis [1]. Also, it is imperative that alternatives be studied in realistic operational settings so that reasonable comparisons of relative performance can be made.

2.4.2 Effectiveness & Cost Analyses

Effectiveness analysis during the conceptual design phase is normally the most complex element of the AoA, oftentimes requiring sophisticated, iterative modeling and simulation (M&S) efforts to capture different system/architectural aspects. There are many different M&S methods available. They range from Markov chains [88, 133], Petri Nets [121, 160], Agent-Based Modeling [35], System Dynamics Models [103], and Discrete-Event Simulation [121], to name a few. SoS architectures often require multiple models to adequately capture the many different modes of behavior that can be exhibited during their operation. Figure 16 provides an excellent overview of the many different disciplines involved in modeling relevant aspects of complex systems.

For M&S to be of value, appropriate Measures of Effectiveness (MoE) and Measures of Performance (MoP) must be carefully chosen. MoEs are qualitative or quantitative measures of a system's performance or characteristic that indicate the degree to which the system performs the task or meets a requirement under specified conditions. MoEs are closely related to mission tasks and objectives so that each alternative can be evaluated against them. MoEs also provide the basis for investigating performance sensitivities to variations of key assumptions and MoP values. Raw quantities like number of units lost or frequency of counter-detection are typically used to express MoEs. MoPs, on the other hand, are typically a measure of a system characteristic. Representative examples include range, velocity, mass, and weapon load-out. MoPs are usually chosen to enable calculation of one or more MoEs [127].

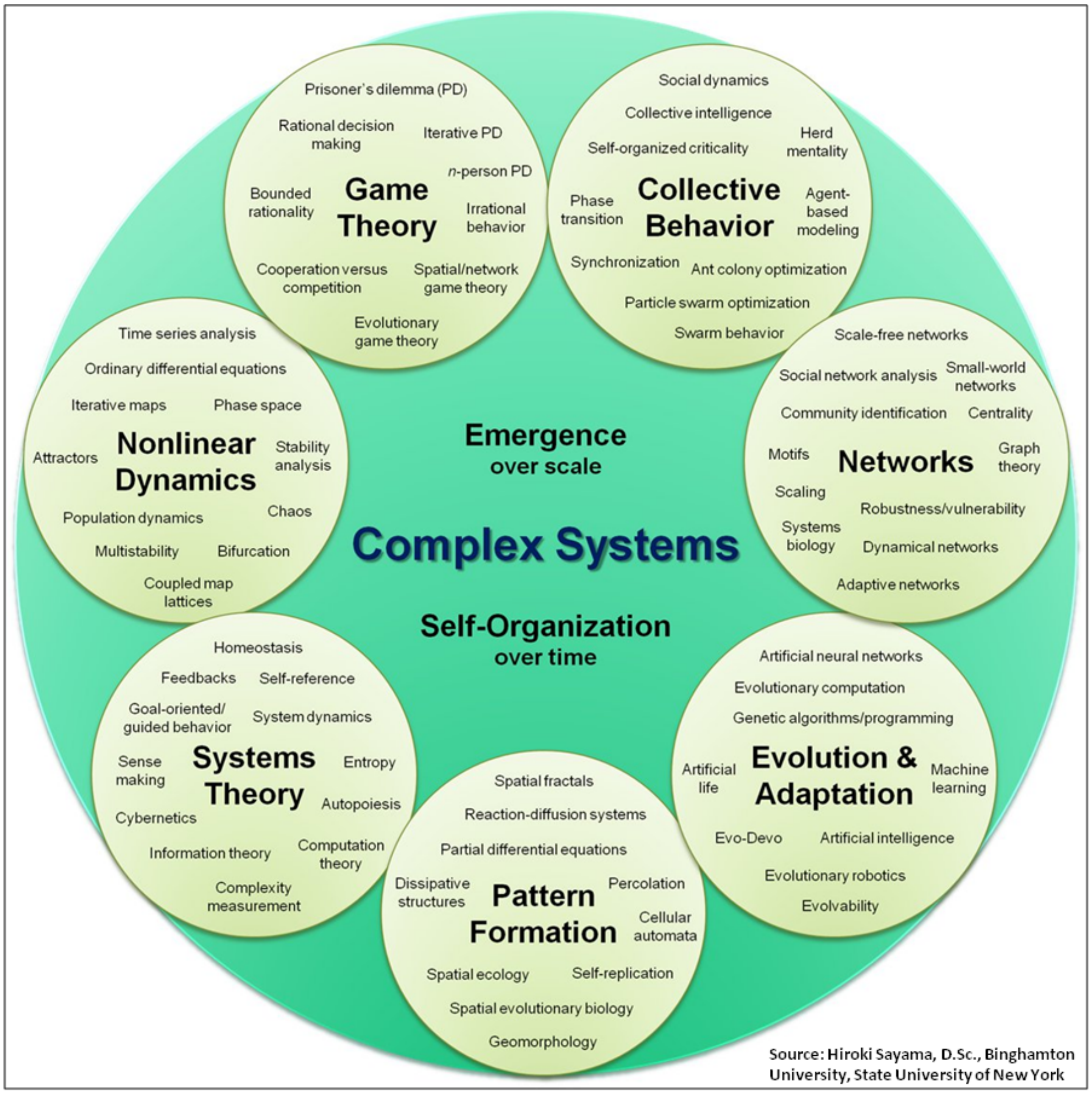


Figure 16: Disciplines Involved in Complex Systems Study.

Cost Analysis is usually conducted in parallel with effectiveness analysis and is considered to be of equal importance. Estimation of the LCC is usually combined with the results of the effectiveness analysis to perform a combined cost-effectiveness comparison. The following is a non-inclusive list of elements usually captured in the LCC estimation [1, 105, 127]:

- Research, Development, Test, and Evaluation Cost (RDT&E)
- Investment Cost (Low Rate Initial Production and Deployment)
- O&S Cost
- Disposal Cost

There are a multitude of cost estimating methodologies and cost models available to the analyst. Examples of different cost methodologies include the following: relying on expert opinion, bottom-up, parametric, and the analogy technique [105, 127]. The bottom-up methodology requires detailed breakdowns and cost data of the various engineering tasks to be completed. In general, this method is more time-consuming. Also, because detailed design data is usually not available in the early stages of design, this method is considered least appropriate for advanced system studies. The analogy method uses actual costs from a similar program and uses adjustments based on the level of difficulty and other differences the new program may have. This method is normally used when there is insufficient data early in the program to use as a basis for a detailed approach. While this method may be applied at any level of detail in the system, it is usually considered too inflexible for trade studies. Lastly, the parametric method is frequently used in the early stages of a program. This particular method involves collecting relevant historical data at an aggregated level of detail and relating it to the area to be estimated through cost estimating relationships (CERs). These CERs are based on actual program cost history, but are at a very high level so that most detail is lost. Parametric cost models are the most appropriate for trade studies and provide the following advantages:

- Less time consuming than traditional bottom-up estimates
- More effective in performing cost trades
- More consistent estimates
- Traceable to the class of systems for which the model is applicable

A major limitation encountered when using a parametric cost model is that its use is applicable only to the range of parameters described in the historical database used for calibration of the CERs. As systems or technologies in use by those systems fall further outside of the database, the validity of the CER lessens. This may also be true as legacy systems are used in new operating environments and scenarios, or are required to interoperate in new ways with other systems. This is an especially important consideration for military SoS. While military SoS may include legacy systems, oftentimes there are new technologies and systems that must be acquired as well. Also, since there is an added emphasis on joint operations, disparate and/or novel components from the different branches of the armed forces may be present to provided the needed capability.

In summary, cost models are usually tailored to a specific discipline or problem domain, such as software or logistics. No cost model can perfectly forecast the future, so there is always some degree of uncertainty, just as with any other type of estimation. A good illustration of this is the "Cone of Uncertainty" developed by the Government Accountability Office's Cost Assessment Guide [85], and reproduced as Figure 17. The Cone of Uncertainty illustrates that cost uncertainty is relatively large in the initial stages of design, but diminishes with progression through the design process. Cost uncertainty then reaches a minimum later in the production and deployment stage of the system life cycle.

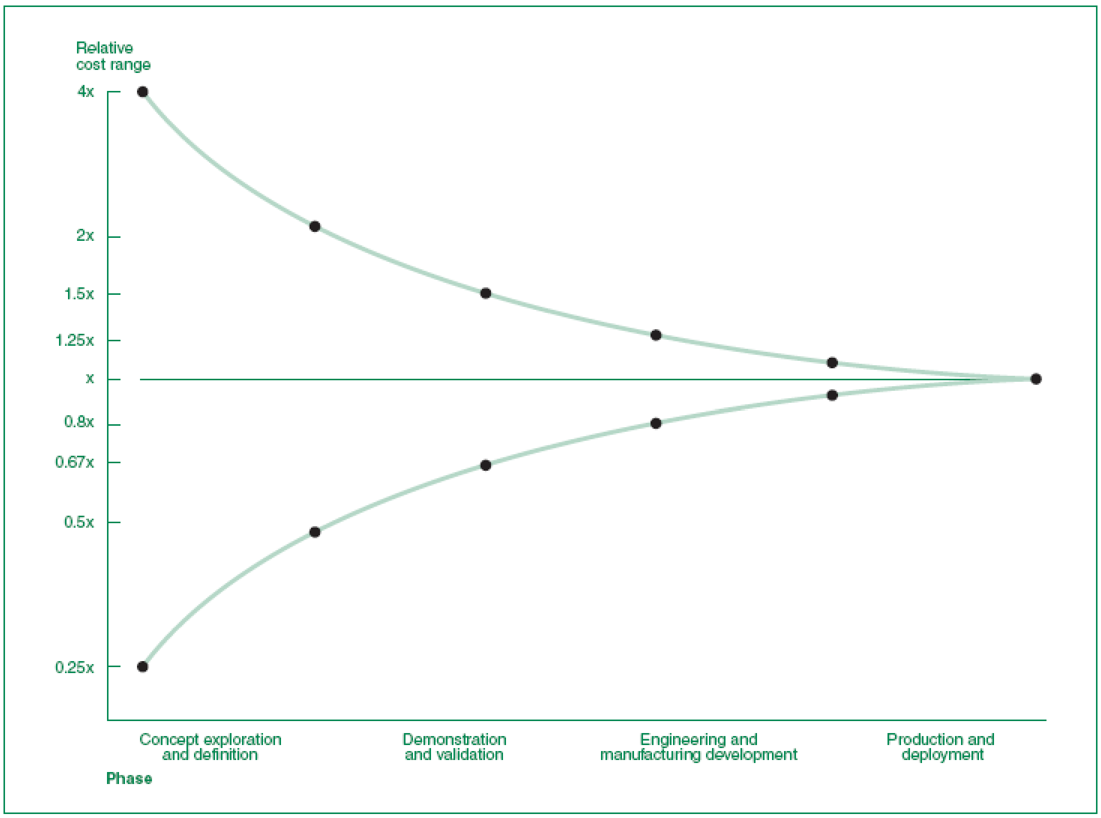


Figure 17: Cost Uncertainty During Different Phases of Acquisition & Design [85].

2.4.3 Cost-Effectiveness Comparisons

The need to assess the value of pursuing an investment alternative or opportunity is not unique to defense acquisition. Firms across various industries must frequently engage in resource allocation decisions when deciding upon investments in new products, services, or R&D efforts. Commercial firms that operate for profit in competitive markets typically have an advantage in that the market communicates the relative value of a good, service, or investment in terms of a monetary price through the relationship of supply and demand. Thus, the commercial firm is better able to objectively compare the costs it will incur as well as the required rate of return it must achieve from an investment of resources when assessing different alternatives. In comparison, it is difficult for the government to determine what the fair market price should be for many of the new defense systems it wishes to acquire. The U.S. government is oftentimes the most important and/or only customer to defense contractors who supply such systems. This results in government acquisition decisions directly impacting the performance of its supplier base and the subsequent distortion of market prices [134]. Augustine provides an insightful summary of the situation [16]:

> On the surface, defense acquisition appears to have little in common with commercial acquisition. For starters, defense acquisition occurs in a monopsony [1]. Further it is replete with mini-monopolies. (From how many places could one have purchased, say, an additional B-2?). Defense acquisition also operates in a governmental system that intentionally traded optimal efficiency for strong checks and balances—such as those implicit in separating the Legislative and Administrative branches. Nonetheless, there are certain fundamentals of sound management which are applicable virtually everywhere, including in the defense acquisition

[1] A *monopsony* occurs where there is imperfect competition, as only one buyer faces many sellers, or the opposite of a monopoly [142]

> process. They are just much more difficult to apply in government, where the stakes are higher, authority less hierarchical, and the spotlight much brighter.

Compounding this problem is the fact that national defense is a service that is broadly consumed for the benefit and welfare of the public. So unlike a good or service sold into a market place by a for-profit firm, there is no direct return on investment that can be measured in terms of dollars and cents. This prompts non-profit organizations and agencies like the government to adopt either Cost-Effectiveness or Cost-Benefit Analysis methods to evaluate different investment alternatives. Cost-Benefit Analysis involves determining the Benefit-to-Cost Ratio (BCR) of a project or alternative. There are several definitions of the BCR, but in general, it can be defined as the ratio of the equivalent worth of benefits to the equivalent worth of costs. The use of the word *worth* to describe both benefits and costs denotes that each is measured in monetary terms [41, 106]. It has been noted that determining equivalent monetary values of some benefits can be a flawed approach in some cases, or not even really possible in others [8, 106]. This proves to be the case for the acquisition of weapons systems in support of national defense. For example, how does one put a precise dollar value on the amount of freedom and safety a B-2 bomber provides? Cost-Benefit Analysis, then, is usually reserved for a limited number of situations, making Cost-Effectiveness Analysis the more commonly utilized methodology of the two [106].

Like Cost-Benefit Analysis, Cost-Effectiveness Analysis also involves determining a ratio. The use of effectiveness instead of benefits means that native units of evaluation can be used. In this case, the Cost-Effectiveness Ratio (CER) merely requires combining cost data with the various MoEs that have been carefully considered and evaluated using M&S. The next step is to eliminate non-viable alternatives. This is followed by a comparative screening focused on selecting the most cost-effective solution(s) for recommendation and development. Rather than use a CER, however, the defense acquisition community traditionally utilizes scatter plots such as the one shown in Figure 18 from the Defense Acquisition Guidebook.

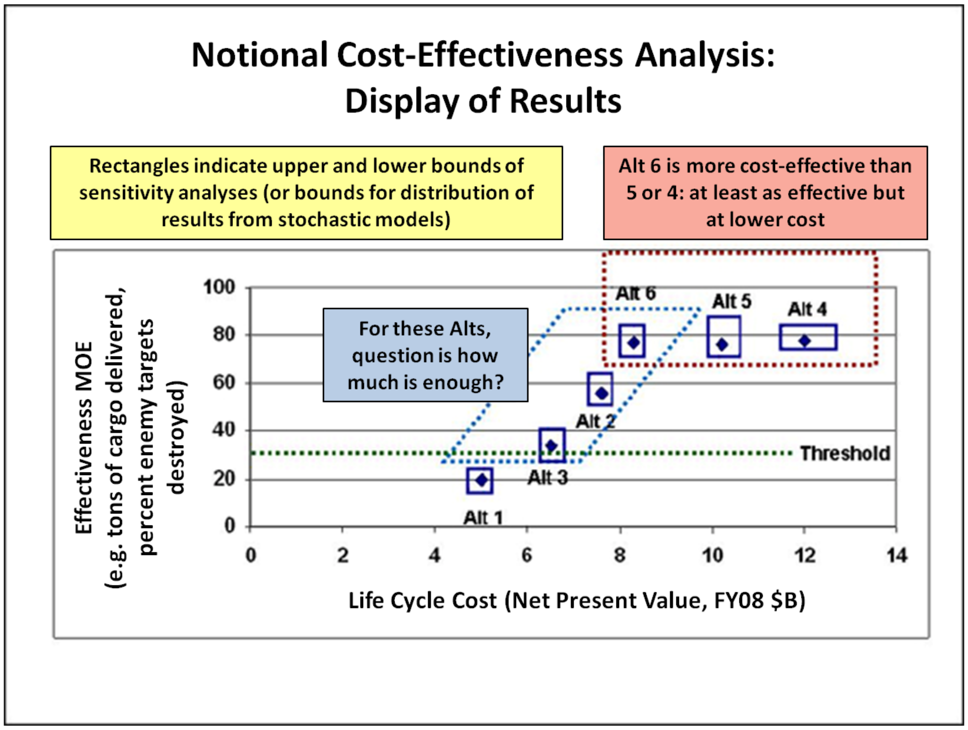


Figure 18: Notional Scatter Plot of Effectiveness vs. Cost [1].

The data is presented in this way to avoid the use of cost-to-effectiveness or effectiveness-to-cost ratios that are more commonly seen in other applications of Cost-Effectiveness Analysis [106]. The rationale for this is provided in the following [1]:

> Note that the notional sample display shown... does not make use of ratios (of effectiveness to cost) for comparing alternatives. Usually, ratios are regarded as potentially misleading because they mask important information. The advantage to the approach in the figure above is that it reduces the original set of alternatives to a small set of viable alternatives for decision makers to consider.

Either implementation of a Cost-Effectiveness Analysis (scatter plot or CER) allows the relative comparison of alternatives for fulfilling a particular set of mission capabilities. Sensitivity analyses are usually included in order to quantify the amount of uncertainty present in the cost and effectiveness estimates and provide an added dimension to decision making. This can be seen in the boxes surrounding each alternative in Figure 18. However, a more complete AoA requires additional criteria that must be taken into consideration. Therefore, a risk assessment must also be conducted, with inputs such as scheduling considerations, technology maturity, and other key programmatic parameters included as well.

From Figure 18, decision makers can safely conclude that Alternative 1 would be a poor selection since it does not meet the minimum required threshold for effectiveness. The same argument could be made for Alternative 3, whose measured uncertainty crosses the threshold boundary. However, the issue is not as clear for the remaining alternatives. Alternative 6 will be chosen over Alternative 2 if the the increase in effectiveness is judged to be worth the additional cost. Also, while Alternative 6 is deemed more cost-effective than Alternatives 4 & 5, this cost-effectiveness may come with some types of programmatic risk not captured in the displayed uncertainty estimates [127]. To further aid the decision maker, the use of multi-attribute portfolio analysis is commonly used to aggregate and display additional useful information [56, 57, 127]. Usually this type of analysis is presented using similar, common visualization formats. These formats are often referred to as portfolio analysis tools (PATs), dashboards, or alternative comparison matrices. Such views are useful for depicting information across multiple capability mission areas. A representative example from the Air Force's Office of Aerospace Studies AoA Handbook is provided as Figure 19 [127]. In Figure 19 the letters R, G, and Y signify the colors Red, Green, and Yellow, respectively.

	Critical						Non-Critical			Risk	Total LCC $(M)
	Mission Task 1			Mission Task 2			Mission Task 3				
	MoE 1-1	MoE 1-2	MoE 1-3	MoE 2-1	MoE 2-2	MoE 2-3	MoE 3-1	MoE 3-2	MoE 3-3		
Alt 1 (baseline)	G	Y	R	G	G	Y	G	R	G	R	$1,200
Alt 2	R	R	G	Y	R	G	G	Y	Y	G	$1,450
Alt 3	Y	G	R	G	Y	Y	Y	G	G	R	$1,457
Alt 4	G	R	G	Y	G	Y	R	G	R	G	$1,786

Figure 19: Notional Matrix of Alternative Comparison Results [127].

Multi-attribute portfolio analysis seeks to provide a top-down depiction of the different alternatives in relation to relevant data that decision makers wish to see, such as performance, cost, and risk. More detailed supporting data is usually provided in different views and made accessible to the decision maker on demand. Different analysis tools usually only vary significantly in their respective features for manipulating the underlying data and the software used for their implementation. Traditional analysis methods such as cost-effectiveness scatter plots and multi-attribute portfolio analysis are helpful to the decision maker. They aid the decision maker in discovering the important tradeoffs that occur between competing alternatives, prompting further analysis and inquiry. Yet the acquisition of weapons programs is fundamentally a resource allocation decision, and these methods still do not address a critical issue that lies at the heart of acquiring military SoS. Levin poignantly addresses this when describing one of the significant limitations of Cost-Effectiveness Analysis [106]:

> That is, we can state whether a given alternative is *relatively* more cost-effective than other alternatives, but we cannot state whether its total benefits exceed its total costs. That can only be ascertained through a cost-benefit analysis.

Keeping in mind the DoD's stated goal of developing affordable weapons systems, the principal conclusion that can be made is that the cost-effectiveness comparison methodology currently in use has room for improvement, especially if applied to more difficult SoS acquisition problems. More specifically, the current cost-effectiveness methodology is adequate for *procurement*, which is defined by the DoD as the act of buying goods and services for the government [38]. According to the DoD, procurement is often (and mistakenly) considered synonymous with acquisition; it is, instead, only one of the many functions performed as part of the acquisition process, which also includes design [38]. For procurement, the cost of complexity has already been converted into monetary costs by progressing through the different stages of design.

Thus, knowing the relative cost-effectiveness of a group of alternatives does not guarantee that the most cost-effective alternative justifies the planned investment of resources such as time, money, or labor. Nor does it help determine whether any additional allocation of resources towards a particular alternative would prove beneficial. Furthermore, the traditional analysis methods in use by the defense acquisition community do not combine cost, schedule, performance, risk, and other programmatic parameters in such a way that decision makers can readily make a balanced assessment across multiple dimensions to judge the *value* of an alternative. Value in this context represents whether or not an alternative is worth an investment of resources in the absolute sense. A cost-benefit analysis would help determine this, but as previously discussed, this proves rather difficult for defense acquisition since benefits must be expressed in monetary terms. Without an objective valuation mechanism for decision makers, they are still left with the task of determining when additional effectiveness is worth additional cost during an AoA [1, 127]. The Air Force's AoA Handbook describes this as a "common AoA dilemma" [127].

The end result is that it may be difficult at first glance to determine which alternatives in the design space hold sufficient promise to be further developed and evolved into more promising solutions for an acceptable level of effort. This can unnecessarily limit design freedom by influencing decision makers to prematurely focus on a handful of alternatives for analysis. Taking all of this into consideration, an improved valuation method for acquiring alternatives that will provide capabilities deemed crucial to national defense is needed. This approach must augment decision making by providing the defense acquisition community with the ability to ascertain when an investment of budgetary, labor, and scheduling resources to acquire systems that will later be included in a complex military SoS architecture is indeed a worthwhile endeavor. To accomplish this during the early stages of acquisition and design, the use of a metric in addition to monetary cost is necessary.

CHAPTER III

RESEARCH FOCUS

3.1 The Impact of Pre-Milestone A Decisions

The social and technical complexities that are a part of acquiring weapons systems for national defense can and have occasionally led to failure. Yet despite the many criticisms and past attempts at reform it is important to remember that the U.S. continues to field arguably the most technologically advanced, capable, and far-reaching military of all time. The defense acquisition process has, in Augustine's words, "provided our armed forces with the equipment that is the envy of the world's military forces. It's just that it could, and should, do even better" [16]. In order to maintain this level of superiority, however, requires constant improvement to the methods and policies currently in place. The sense of urgency accompanying these improvements grows as the threat environment itself evolves, forcing the systems operating within that environment to evolve as well.

This research seeks to address the technical challenges of refining capabilities-based SoSSE practices by researching methods to deal with evaluating the growing complexities of SoS architectures that inevitably impact decision making. As a reminder, Milestones A, B, and C are where critical management decisions are made when acquiring weapons systems. Milestone A encompasses the conceptual design effort. This is where system architecting is usually initiated and where concept refinement takes place. It is here that the MDA can approve a materiel development decision and grant formal entry into the acquisition process. Milestone B marks formal acquisition program initiation with dedicated funding, and Milestone C is where the MDA deems a system/technology mature enough commit to low-rate initial production. The importance of using a capabilities-based approach and applying sound SoSSE practices from the very beginning is made readily apparent in Figure 20.

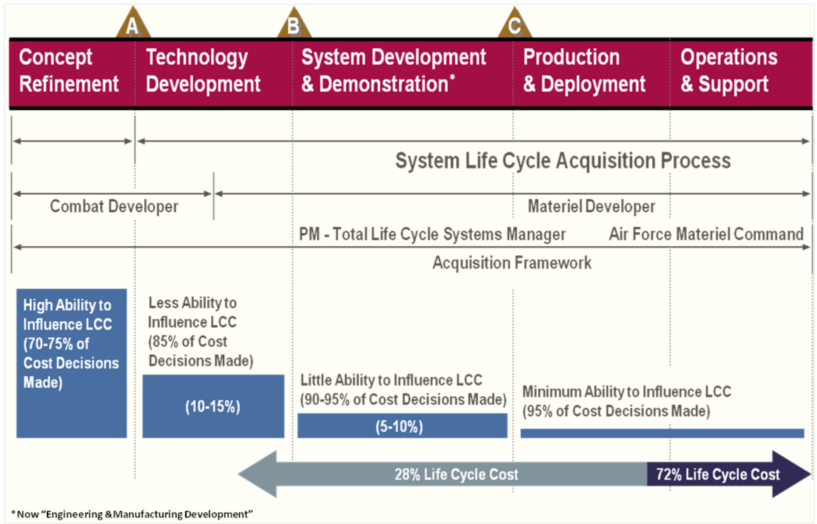


Figure 20: Life Cycle Cost at Different Acquisition Milestones [14].

Figure 20 shows that between 70%–75% of the LCC is influenced by decisions made prior to Milestone A, while approximately 72% of the funding is not actually expended until after Milestone C has been passed. It is important to note that these decisions not only impact cost, but also schedule and performance. A similar trend is also observed for computer software architectures, which the DoD relies on heavily as well. For software architectures, often less than 1% of the total cost can be attributed to system architecting related efforts while those decisions define up to 80% of the total development cost [168]. The conclusion to be drawn here is that while the application of sound SoSSE methods and practices are needed throughout the system life cycle, they are most critical and will have the most impact on cost, schedule, and performance during the pre-Milestone A phase of the JCIDS process [10]. The problem that arises, though, is that the early stages of conceptual design is the point at which there is the greatest amount of design flexibility, but in contrast the least amount of design knowledge [117, 165]. As the design process progresses from the conceptual stage to more detailed phases such as manufacturing, production, and O&S, more knowledge is gained. However, this knowledge comes at the expense of restricting design freedom to the point where changes quickly become more costly and burdensome. Thus, the focus of this research will be applied to supporting decisions made during the pre-Milestone A phase. Specifically, the focus will be on aiding decision makers in the selection of cost-effective complex military SoS architectures during the AoA that have the greatest likelihood for successful acquisition.

3.2 Seeds of Failure Planted During Pre-Milestone A

A 2008 report delivered to the Air Force by the National Research Council of the National Academies was commissioned by the Deputy Assistant Secretary of the Air Force for Science, Technology, and Engineering. The main goal of the study was to "examine the role that systems engineering can play during the defense acquisition life cycle in addressing the root causes of program failure, especially during pre-Milestone A and the early phases of a program" [10]. The 2008 Pre-Milestone A study outlines 6 drivers of cost, development time, and performance risk addressable by SE processes. These "Six Seeds of Failure" are shown in Figure 21.

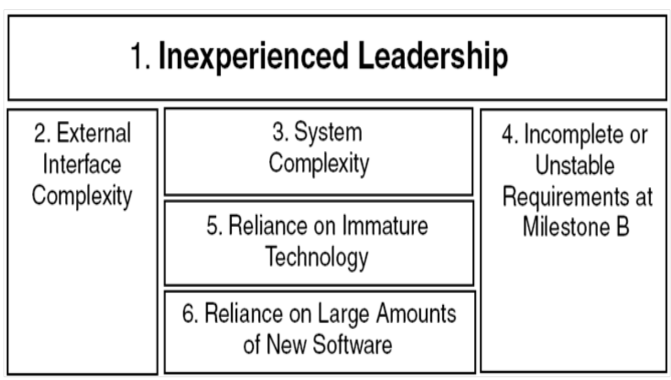


Figure 21: The Six Seeds of Failure During Pre-Milestone A and B [10].

The first seed of failure identified, inexperienced leadership, considers both the length of experience and the number of programs that systems engineering/acquisition professionals have participated in. Obviously, the possible solutions to mitigate inexperienced leadership will fall far from being exclusively technical ones. As the study points out, however, the main contribution that the focus of this research can provide will be in correctly identifying the risk or difficulty associated with each alternative. In this way, the most experienced personnel can be allocated to projects deemed worth pursuing, but that are considered to be high risk. To accomplish this, the contribution to program risk from the other seeds of failure must be reliably assessed as well. The study notes, for example, that "the use of unproven technology in large system developments can introduce a high risk of schedule and cost growth." Likewise, large and complex software elements have historically been the source of high costs and long development times on many programs as well. The Pre-Milestone A study also notes that it is not unusual for programs to proceed beyond Milestone A with incomplete or unstable requirements. The study provides a basic explanation of how this trend came to be:

> One characteristic of very complex system developments during World War II and the early Cold War years was the simplicity and urgency of the needs and missions. Beating the Germans to the atomic bomb, the Russians to the Moon, penetrating the Iron Curtain, and so on, provided clear, urgent goals that galvanized the sponsors of the complex systems and focused and empowered the government and contractor teams. Such clear, driving missions, and the simple user interfaces that they required, allowed the program team to develop its concepts quickly and to keep the top-level requirements stable until IOC.
>
> In the post-Cold War era, the immediacy of the threats often seems less apparent, and programs often try to serve many missions and users with a single system, or system of systems. The interaction of multiple systems that were not designed together (*e.g.*, military satellite communications

> [MILSATCOM]), often termed 'systems of systems,' also can greatly increase the difficulty of creating a stable requirements base for a new system.

The Pre-Milestone A Study acknowledges that funding instability, while the result of a political process and not systems engineering, is another driver of unstable requirements that affects cost, schedule, and performance. It goes on to suggest placing greater emphasis on making schedule a key performance parameter. This sentiment is mirrored in the 2006 Defense Acquisition Performance Assessment Report to the Deputy Secretary of Defense [16]. Both sets of findings strongly advocate considering the "time value of a capability" in order to seek concepts that can deliver initial capability either within about 5 years from Milestone B or no more than six years from Milestone A [10, 16]. Since the time value of a capability is typically used later during the AoA process to down-select to a smaller group of alternatives, decision makers may not afford it the attention usually attributed to cost and effectiveness analyses. However, the time value of a capability necessarily influences cost, schedule, and performance tradeoffs. For example, an 80% solution developed earlier may be preferred to a 100% solution delivered significantly later. The Pre-Milestone A Study goes on to say that "Further, extended delivery times run the risk of the system becoming obsolete before deployment and can be an indication that the concept is excessively complex or excessively dependent on immature technology in its first delivery." A more favorable method is an evolutionary acquisition strategy, where a capability that is initially fielded can be upgraded over time as technologies mature and operational requirements become clearer. This is usually the case with past successful acquisition programs, case-in-point being the F-16 Fighting Falcon [16].

The two remaining seeds are focused on complexity, making it a key design issue. As Maier succinctly states, "Generally speaking, the more complex a system, the more difficult it is to design, build, and use" and "When architects and builders are asked to explain cost overruns and schedule delays, by far the most common, and quite valid, explanation is that the system is much more complex than originally thought" [113]. The Pre-Milestone A Study describes some of the relevant factors that have contributed to overall system complexity, and also provides recommendations on limiting complexity in system design:

- External Interface Complexity: The concept of network-centric operations, in particular, can introduce external complexity. The complex processes necessary to coordinate these communities of interest seem to have hidden costs that can add many years to the development cycle and lead to substantial budget overruns. In addition, systems dependent on highly complex external interfaces can be far more difficult to operate after deployment. SE should treat the minimization of external interface complexity as a key driver in selecting concepts and architecture. Simplifying and standardizing the ways that external users access the system and seeking to minimize the degree to which the systems capabilities must be tailored specifically for individual users can help.

- System Complexity: The flexibility and capability enabled by advances in electronics technology and software provide systems designers with far more options than their predecessors enjoyed 30 years ago. The downside is that these new capabilities can tempt designers into unnecessarily complex concepts and designs that impose a "cost of internal complexity" similar to the external complexity costs described above. SE should treat complexity minimization as a key driver in selecting concepts and architecture. Architecture selection can have a powerful effect on controlling complexity.

The trend towards increasing system complexity is an ongoing study conducted by the Defense Advanced Research Projects Agency's (DARPA) META program [77]. Figure 22 presents META's initial findings.

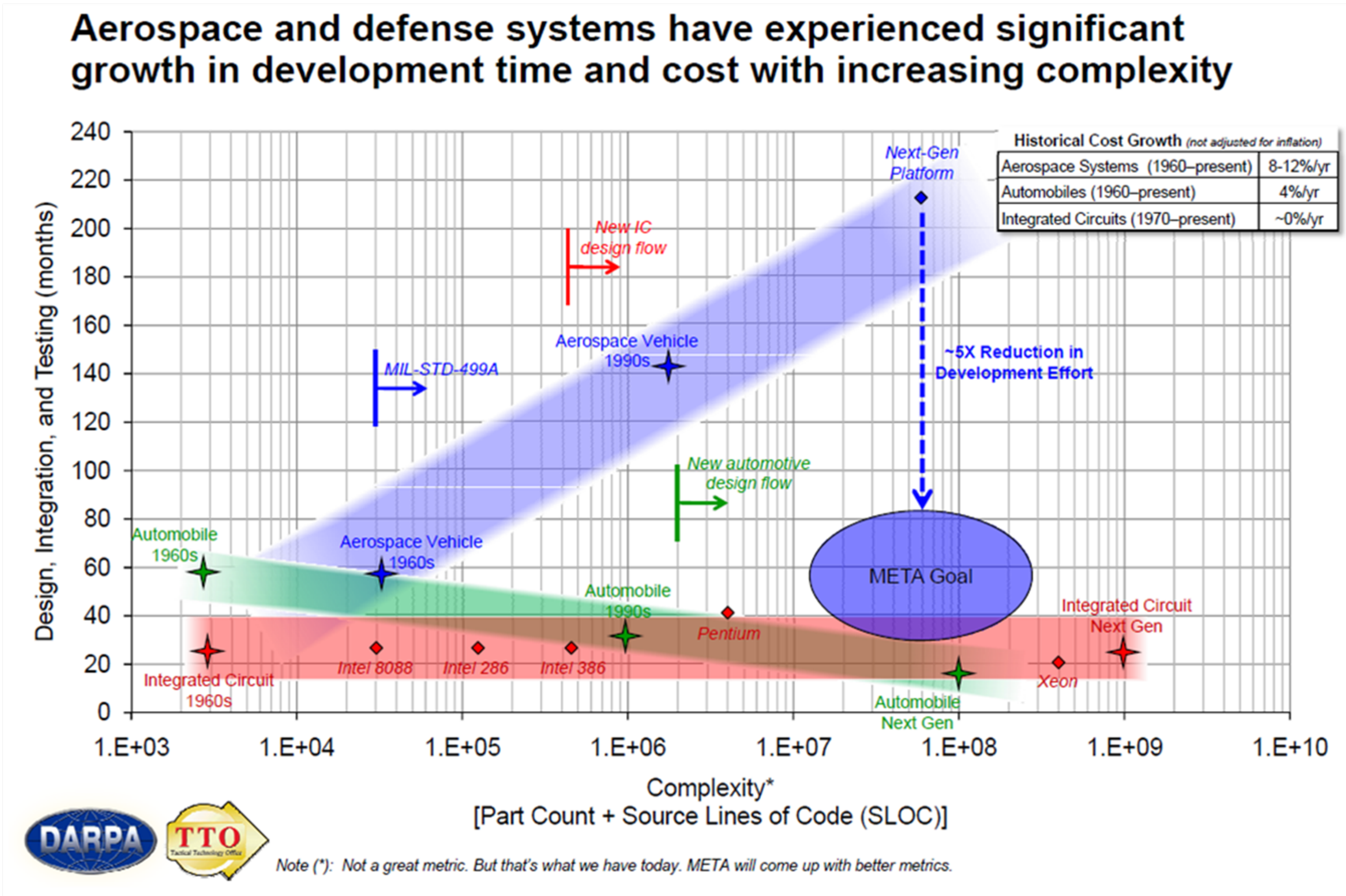


Figure 22: DARPA Assessment of the Impact of Complexity on Cost & Schedule Growth [77].

Figure 22 highlights the correlation between perceived complexity and the impact on both cost and schedule for three different types of systems. Figure 22 clearly shows that while the complexity of automobiles and integrated circuits have increased over time, there has been significantly less cost and schedule growth for these types of systems in comparison to the cost and schedule growth experienced by aerospace systems. The complexity metric utilized by META is a linear combination of part count and source lines of code. META acknowledges, however, that this metric needs improvement and intends to develop a more suitable metric. In comparison, a complexity-based risk assessment (CoBRA) method developed by Bearden uses a complexity index to show the impact of complexity on cost and schedule during satellite design [7, 26, 27]. This is shown in Figure 23.

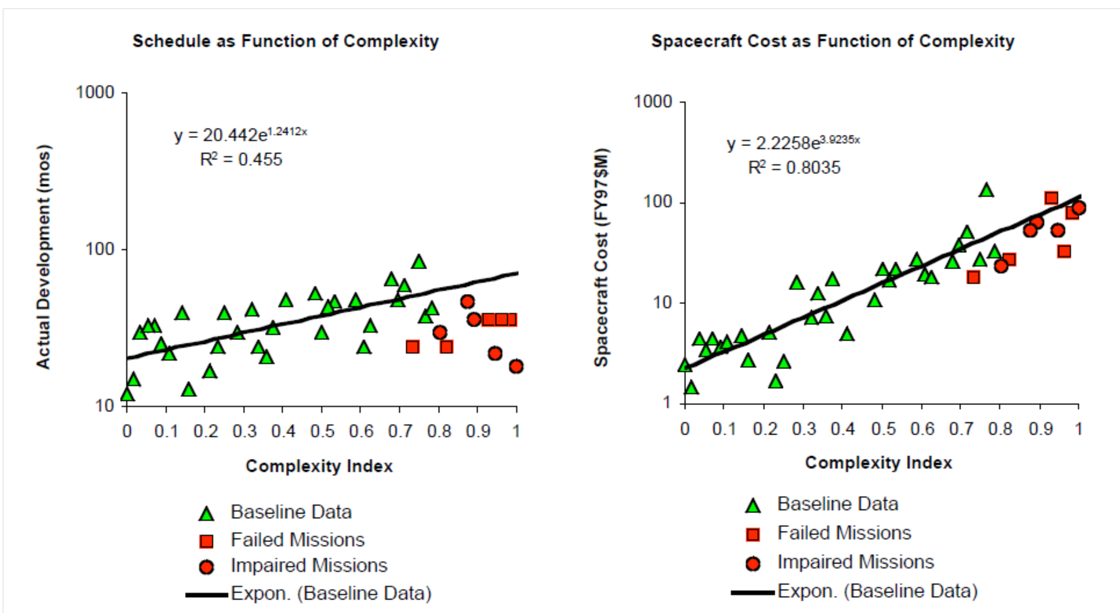


Figure 23: Cost & Schedule as a Function of Complexity for NASA Planetary & Earth-Orbiting Missions [26].

CoBRA is an analogy based estimating technique that relies on information from planetary and earth-orbiting satellite missions from the National Aeronautics & Space Administration (NASA) historical databases. The complexity index is calculated using 21 separate subsystem parameters such as satellite launch mass, design life, solar array area, and propulsion type. The parameters are selected based on requirements that exert a significant influence on spacecraft design. The minimum, maximum, and average value for each parameter is evaluated by analyzing the historical database of previous missions. These results are then aggregated into a normalized complexity index. A proposed mission is given a complexity index close to 100% if the values of many of its subsystem parameters lie near values that historically correlate to greater cost and schedule overruns. It follows that, "The expectation is that a proposed mission is on the road to success if the locus of the cost (and schedule) versus complexity point lies in the vicinity of the data for successful missions in the past" [7]. One of Bearden's key findings is that "A comparison of NASA planetary and earth-orbiting missions showed that low-cost planetary missions cost more, are developed faster, and fail more often than do earth-orbiting missisions" [26]. This assessment further emphasizes the impact of complexity on cost, schedule, and performance tradeoffs. While this method of estimating complexity is more sophisticated than a simple part count, Bearden acknowledges that the relative importance of the parameters contributing to the complexity index and correlation among them in representing and/or driving technology selections remains to be investigated [26]. Since the complexity index relies on a historical database, this method may be unsuitable when considering new technologies or systems that trend away from the historical database.

An example of a parametric SE cost model that includes the effects of architecture complexity is the Constructive Systems Engineering Cost Model (COSYSMO) developed at the University of Southern California Center for Software Engineering. COSYSMO is used to estimate the time and effort associated with performing the system engineering task for large scale hardware and/or software projects [159]. COSYSMO's CER is as follows:

$$PM_{NS} = A \times (Size)^E \times \prod_{i=1}^{n} EM_i \tag{2}$$

Where:

- PM_{NS} = estimated effort in Person Months [1] based on the nominal schedule (response variable)
- A = calibration constant derived from historical project data
- Size = determined by computing the weighted average of the size drivers
- E = exponent representing the (dis)economy of scale dependent on size drivers
- n = number of cost drivers (12)
- EM = effort multiplier for the i^{th} cost driver. The geometric product results in an overall effort adjustment factor to the nominal effort.

The key inputs to the CER are the calibration constant and the size and cost drivers. The calibration constant is derived from over 50 projects provided by major aerospace and defense companies such as Raytheon, Northrup Grumman, Lockheed Martin, SAIC, General Dynamics, and BAE systems. The following is a brief description of the four COSYSMO size drivers:

1. Number of System Requirements: The number of requirements taken from the system specification. A requirement is a statement of capability or attribute containing a normative verb such as shall or will. It may be functional or system service-oriented in nature depending on the methodology used for specification. System requirements can typically be quantified by counting the number of applicable shall's or will's in the system or marketing specification.

[1] Person Months is a man-hour estimate [3] used especially on large projects as a basis for cost accounting and wages.

2. Number of Major Interfaces: The number of shared major physical and logical boundaries between system components or functions (internal interfaces) and those external to the system (external interfaces). These interfaces typically can be quantified by counting the number of interfaces identified in either the system's context diagram and/or by counting the significant interfaces in applicable Interface Control Documents.

3. Number of Operational Scenarios: The number of operational scenarios that a system is specified to satisfy. Such threads typically result in end-to-end test scenarios that are developed to validate the system satisfies its requirements. The number of scenarios can typically be quantified by counting the number of end-to-end tests used to validate the system functionality and performance. They can also be calculated by counting the number of high-level use cases developed as part of the operational architecture.

4. Number of Unique Algorithms: The number of newly defined or significantly altered functions that require unique mathematical algorithms to be derived in order to achieve the system performance requirements.

The 12 COSYSMO cost drivers [1], or effort multipliers, are divided into 5 Application Factors and 7 Team Factors. While COSYSMO provides a comprehensive formulation of key SE factors in its cost estimation, many of the cost drivers are based on a qualitative scale. This includes architecture complexity, which COSYSMO defines as the relative difficulty of determining and managing the system architecture in terms of platforms, standards, components, connectors (protocols), and constraints. A listing of the cost drivers is presented in Table 1.

[1] Level of Service Requirements measures the difficulty and criticality of satisfying Key Performance Parameters such as security, safety, and response time. Migration Complexity is the difficulty in migrating the system from previous system components, databases, or work flows for example. For further details see Reference [159].

Table 1: COSYSMO Cost Drivers.

Team Factors	Application Factors
Stakeholder Team Cohesion	Requirements Understanding
Personnel Capability	Architecture Complexity
Personnel Experience/Continuity	Level of Service Requirements
Process Maturity	Migration Complexity
Multi-site Coordination	Technology Maturity
Formality of Deliverables	
Tool Support	

In comparison, Technomics provides a simpler parametric cost model for estimating RDT&E costs [98]. The Technomics model relies on information captured in DoDAF models and explores the interdependencies that exist between system elements as a result of their interconnections. Therefore, the system architecture is described in terms of nodes and links. A node is described as an element of architecture that produces, consumes, or processes data. Thus, there are three node types: N_s is a node that sends information, N_r is a node that receives information, and $N_{s/r}$ is a node that both sends and receives information. A link is a representation of the physical realization of connectivity between Nodes. A link can either allow uni-directional information flow or bi-directional information flow. The total number of links is expressed as L_t, while the total number of nodes is N_t. The cost model uses the following equations:

$$RDT\&E\$ = aN_e^b \tag{3}$$

$$N_e = \left(dN_{s/r} + gN_s + hN_r\right)\left(\frac{\frac{L_t}{N_t}}{avg\left(\frac{L_t}{N_t}\right)}\right)^c \tag{4}$$

The first term in parentheses in Equation (4) captures the complexity associated with the types of nodes. The second term in parentheses captures the connectivity complexity associated with the system. The parameters a, b, d, g, h, and c used in both equations are estimated using residual-minimization techniques. The average value of L_t/N_t is obtained from the database of all architectures under evaluation. An analysis using the Technomics model leads one to the conclusion that understanding the relationship between links and nodes is critical in understanding the potential for RDT&E cost growth [98].

In summary, each of the four analyses previously discussed seek to capture the impact of complexity on system design. While this impact is often cast in the negative light of cost and schedule overruns, one must also bear in mind that complexity is not inherently a bad property. As Moses explains, "you usually have to expend complexity dollars to achieve useful goals, such as increased functionality, efficiency, or flexibility" [153]. Since added complexity often carries with it additional risk, this makes it necessary to determine what level of complexity is appropriate and how that complexity should be managed. Therefore, while complexity can add value in achieving capabilities, complexity and affordability are often two conflicting objectives that need to be traded against each other to find a compromise. As previously mentioned, the Pre-Milestone A Study emphasizes that architecture selection can have a powerful effect. Then it stands to reason that the complexity that results when systems interoperate as part of a SoS architecture should be given greater consideration upfront in the decision-making process [53]. In this way, an accurate measure of complexity can serve as an even more powerful tool to decision makers during the AoA.

3.3 Research Objective

The objective of this research is to develop a valuation methodology suitable for acquisition-level decision making during the pre-Milestone A phase of acquisition. To determine how well this method aids decision makers in the down-selection of alternatives for the implementation of a cost-effective, evolutionary acquisition strategy, the valuation method should be judged according to a list of criteria that represents the stated needs of the acquisition community. Therefore, the following should be addressed:

1. Must take into account programmatic risks, uncertainties, and sensitivities [1, 127]

2. Must take into account the hidden costs of complexity [10]

3. Must consider the future direction of the SoS in the developing architecture, *i.e.* how the architecture will evolve [69]

4. Must take into account the "time value of capability" [10, 16]

5. Must help determine when additional effectiveness is worth an additional investment of resources when comparing alternatives [1, 127]

6. Must ensure that the advantages and disadvantages of each alternative are presented in a clear and unbiased manner, and that it depicts the analysis results, understandable interpretations, and defensible recommendations [127]

Meeting the research objective would not be possible without leveraging past and current advancements from the following disciplines:

- The development of architecture frameworks for organizing and describing key SoS attributes and information

- Advancements in the emerging field of Complexity Science for characterizing and measuring complexity

- Previous and newly emerging methods for applying financial valuation methods to business and engineering problems for decision making

- Advancements in computing and IT that allow for the efficient manipulation, display, and comprehension of data and information

- The development of graph/network theory as a visual and mathematical aid in analyzing complex architectures

Currently, estimates of system and architecture complexity are limited in their use. Complexity is merely being used as a parameter to further inform cost and schedule forecasts. *For this research, however, the treatment of complexity as the primary focus during pre-Milestone A decision making will be the key difference that sets this method apart from current practices. More specifically, since complexity is already thought of as a type of cost that is exacted for increased functionality [16, 153], this research advocates that it can be similarly treated as such during the early phases of design.* Applying this conceptual shift in order to "monetize complexity" means that now, more advanced financial theories and methods can be brought to bear during the AoA to gauge the true value of competing alternatives. For example, it may be possible to apply a discounting mechanism using an appropriate valuation framework that takes into account attempts to manage complexity with programmatic resources such as budget, labor, and scheduling. The hope is that this will enable better decision making.

CHAPTER IV

RESEARCH QUESTIONS

4.1 Research Question #1: Measuring SoS Architecture Complexity

A summary of the observations leading to the first research question are:

O1a. Defense acquisition is increasingly more complex in a network driven, joint capability focused acquisition environment. In order for weapons systems to deliver greater war fighting capabilities, military SoS must be considered in the context of design. This necessitates improved SoSSE methods and architecting principles to manage and evolve these complex military SoS.

O1b. Complexity is a key design driver that impacts the cost, schedule, and performance of acquisition programs. Failure to adequately account for complexity in pre-Milestone A acquisition negatively impacts the timely acquisition of cost-effective systems to meet warfighter needs.

O1c. Varied methods for measuring complexity exist. Different complexity measures used by DARPA's META program, CoBRA, COSYSMO, and Technomics, for example, range from qualitative to more quantitative assessments of complexity.

Based on observations O1a.—O1c., the focus of the first research question becomes:

RQ1. What is an appropriate method of measuring military SoS complexity during the pre-Milestone A phase of acquisition to aid decision makers in architecture selection?

4.2 Research Question #2: Developing a Valuation Framework

A summary of the observations leading to the second research question are:

O2a. Traditional analysis methods such as cost-effectiveness scatter plots and multi-attribute PATs are helpful to the decision maker. Yet, these methods still leave decision makers with the task of determining when additional effectiveness is worth an additional investment of resources when comparing alternatives. An improved valuation methodology is needed.

O2b. Complexity can be considered as the cost inherent in design for achieving increased functionality, efficiency, or flexibility. However, current methods of incorporating architecture complexity into an AoA are limited in application.

O2c. The conceptual shift of monetizing complexity, or treating complexity as a cost allows for more advanced financial theories and methods to be used, such as applying a discounting mechanism directly to measured architecture complexity.

Based on observations O2a.—O2c., the second research question is as follows:

RQ2. What financial theories and methods are appropriate for developing a complexity-based valuation method to determine when additional effectiveness is worth additional resources when performing an analysis of alternatives?

CHAPTER V

MEASURING ARCHITECTURE COMPLEXITY

5.1 Defining a Complex System

Churchman & Ratoosh make clear that, "it is meaningless to speak of measurement unless there is already available some form of definition" [48]. So before evaluating or deriving a complexity measure, complexity must first be defined in terms readily applicable to complex systems and their behavior. To do this, an understanding of what is meant by a *complex system* must be obtained. Then it will be possible to infer possible architectural and design characteristics that give rise to complex phenomena. Complexity itself is a concept that impacts numerous disciplines, from biology to economics to mathematics, computer science, and IT. As a result, attempts to define and characterize complexity span many different approaches.

The Oxford dictionary meaning of the word *complex* is "consisting of many different and connected parts" or "not easy to analyse or understand; complicated or intricate" [4]. The word complex can be traced back to the Latin *complexus*, which means 'plaited', or braided together and intertwined [4]. Though the use of the word *complicated* is often used synonymously with the word *complex*, it has been suggested by some that understanding and engineering complex systems represents a new design paradigm [36, 151]. In this new paradigm both Complexity Science and system architecting play prominent roles. At its core, the nascent field of Complexity Science centers on the philosophical debate between Reductionism and Holism [149] to distinguish the complex from that which is merely complicated. Each philosophy seeks to define whether the properties and behavior of systems can best be understood by studying the fundamental components individually or only rather by holistically observing the interactions of components as an entire, functioning system. Complexity Science favors the latter. Whereas a system that is merely complicated may have many parts arranged in intricate ways, its behavior can still be well understood from a careful decomposition of its parts. Former NASA administrator Michael Griffin provides a succinct summary when he states, “Complicated is decomposable, which is what systems engineering is based on. Complex systems are no longer strictly decomposable, and systems engineering has to adapt” [161]. With this in mind, two pertinent definitions describing a complex system are:

1) A system composed of interconnected parts that as a whole exhibit one or more properties (behavior among the possible properties) not obvious from the properties of the individual parts [99].

2) A system having many interrelated, interconnected, or interwoven elements and interfaces [52].

Examples of non-obvious behavior that are typically attributed to complex systems include adaptive behavior, self-organization, and the difficult-to-predict interactions between elements of a system. Global emergence is also observed, and Bar-Yam describes this as collective behavior that is contained in the behavior of parts only if they are studied in the context in which they are found [22]. Also, complex systems tend to evolve desirable traits such as robustness and adaptability [153].

In general, these behaviors and traits are a direct result of the manner in which the functionality of a system, or group of systems, is achieved. A function can be defined as a necessary task, action, or activity that must be accomplished. Functions are performed or accomplished through the use of a combination of resources such as information, equipment, personnel, facilities, or software [65]. For a system or SoS composed of interconnected parts, these resources are transmitted via interfaces.

This being the case, it follows that a careful examination of a system in terms of both functionality and resource sharing is needed to fully understand a system's complexity, or lack thereof.

5.1.1 Complex System Classification

5.1.1.1 Physical/Functional Relationships

Balestrini-Robinson provides a relevant classification scheme for different system types that is based on the relationship between a system's observable physical and functional traits. Figure 24, along with the accompanying descriptions, summarizes this classification scheme.

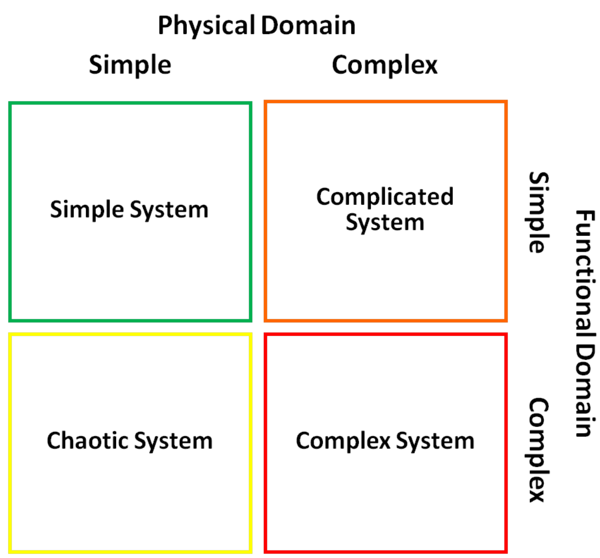


Figure 24: Complex System Classification Using Physical-to-Functional Relationships [21].

- Simple System: Simple in both the physical and functional domains. The system is predictable and traditional engineering techniques can be successfully applied.

- Complicated System: Physically complex, yet functionally simple. These systems are less predictable than those classified as simple because the functional requirements are satisfied, but cannot be ensured under all possible conditions and states. A reductionist approach still applies.

- Non-deterministic Chaotic System: Functionally complex, yet physically simple. The non-deterministic system is influenced by random perturbations to produce the appearance of complexity. This behavior, while similar in appearance to chaos, has rather different implications for prediction and control. These systems have the property of being predictable for short times, yet completely unpredictable over long time periods [70]. Solved using traditional Robust Design Techniques [116].

- Complex System: Both physically and functionally complex. Emergent behavior and highly nonlinear interactions make predicting and understanding these systems difficult. Must architect the system to behave correctly by tailoring the emergent behaviors.

Figure 25 provides a visual example of the behavior of each type of system in terms of a time-series [21]. This illustrates the system state regimes where the transition from static, linear behavior to nonlinear patterns of behavior occur. Note that the complex regime lies between the periodic and chaotic system states. Balestrini-Robinson makes note that "The difficulty in determining where exactly the system is 'complex' is evident from the picture, unlike the periodic and static states, the chaotic and complex are difficult to discern by the human mind, *i.e.*, there is not a clear boundary where the system is complex, or chaotic" [21].

5.1.1.2 Nonlinearity & Coupling

Perrow, in his book *Normal Accidents*, provides a complementary means of classifying complex systems in terms of system interactions and coupling. Taking into account both couplings and interactions, Perrow illustrates how an alternative mapping of system complexity emerges. This is depicted in Figure 26.

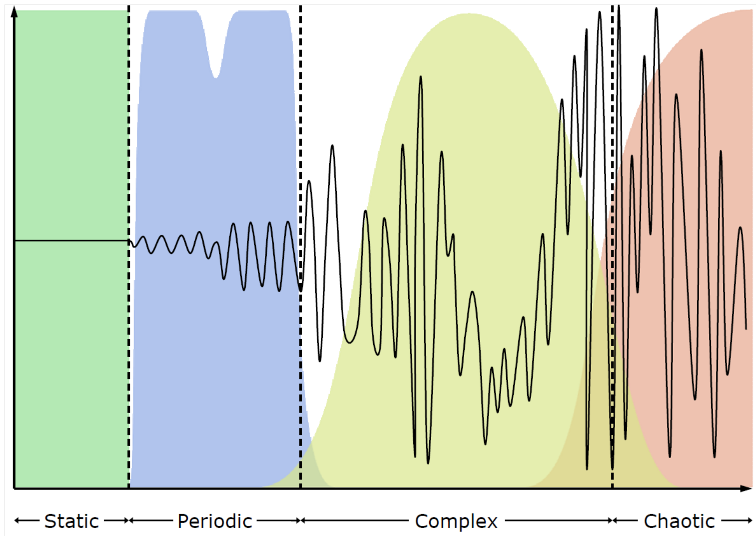


Figure 25: System State Regimes [21].

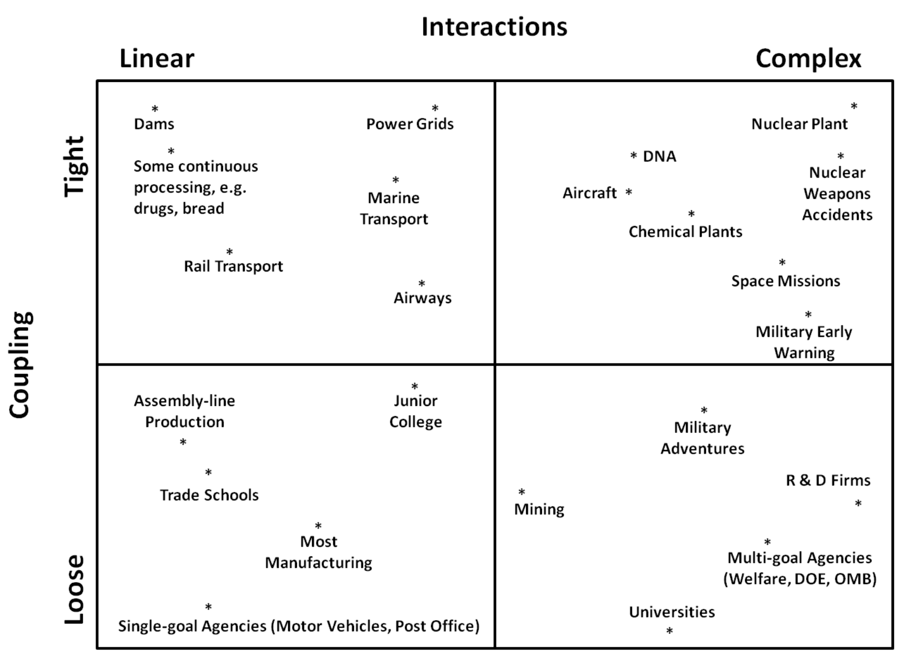


Figure 26: Perrow's Interaction/Coupling Chart [136].

Perrow defines *Linear interactions* as those interactions that occur within the system that are usually expected and familiar, or at the very least visible even if unplanned; linear interactions are readily comprehensible. In opposition, "*Complex interactions* are those of unfamiliar sequences, or unplanned and unexpected sequences, and either not visible or not immediately comprehensible" [136]. Perrow also explores the effect on complexity due to coupling in the design and function of a system. Perrow explains, "tightly coupled systems have more time-dependent processes: they cannot wait or stand by until attended to" [136]. Thus, tightly coupled systems have little slack, buffers and redundancies must be designed in, and functional sequencing remains invariant. Loosely coupled systems, by contrast, can tolerate delays and can remain in "standby mode" if necessary. Additionally, the order of functional sequences can be changed and buffers and redundancies are "fortuitously available" [136]. These are but a few of the differences in tendencies between tightly coupled and loosely coupled systems according to Perrow. Furthermore, Perrow provides an analysis of the different system groupings:

> By combining our two variables in this way, a number of conclusions can be made. First, it is clear that the two variables are largely independent. Examine the top of the chart from left to right. Dams, power grids, and nuclear plants are all roughly on the same line, indicating a similar degree of tight coupling. But they differ greatly on the interaction variable. While there are few unexpected interactions possible in dams, and not that many in power grids, there are many in nuclear plants.

Perrow goes on to state the following regarding systems with loose coupling:

> The post office and the university, then, are similar with regard to coupling; both can recover from upsets because sequences are not that inevitable, there is slack in resources, substitutions are possible. But the

> post office is largely limited to linear interactions, while the university is full of many potentially complex ones that can reach unexpectedly into other parts of the system.

In developing the Interaction/Coupling Chart, Perrow acknowledges, however, that "the degree of coupling and type of interactions have been inferred from a rough idea of the frequency of system accidents in the various systems, rather than derived from analysis of the properties of the systems independent of the nature of their failures" [136]. This highlights the need for an objective quantification of independent, observable system properties that contribute to overall system complexity. Nevertheless, from each of these analyses, it is readily apparent that many engineered systems are considered complex due to the couplings and interactions between system components that are oftentimes considered physically and functionally complex in their own right [21]. Naturally occurring biological systems, on the other hand, show us that complex collective behavior can also arise from networks of individual components with little or no centralized control. This complex collective behavior occurs even when the individual components are relatively simple in terms of structure, behavior, and variation [123]. This provides an added third dimension to complexity to what was previously shown in Figure 24. Relevant examples of these system types include insect colonies such as ants and bees, the collection of neurons in the human brain, and human immune systems. The self-organization exhibited by these systems appears to be common in nature and can be produced by simple processes operating locally on simple agents or components [36]. In terms of engineered systems, this type of complexity can be seen most often in distributed computing architectures, where multiple autonomous computers (or even multiple parallel processors on one or more computer) communicate through a network to fulfill tasking. In other fields such as satellite design, recent emphasis has been placed on fractionated architectures, where smaller and cheaper space modules are wirelessly interconnected to form a “virtual satellite” [6]. DARPA’s F6 Program is a recent example of this kind of system architecture, and a conceptual graphic is shown in Figure 27 [6, 39].

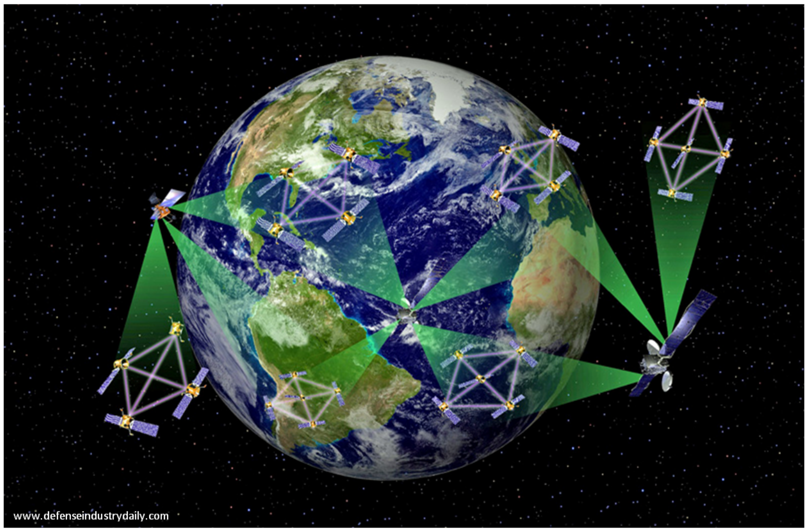


Figure 27: DARPA F6 Fractionated Satellite Architecture Concept.

Such systems, whether naturally occurring or engineered, highlight the importance of including information processing and computation when describing complexity [163]. The implication for engineered systems is that even when components are functionally simpler and more homogeneous than competing architectures, extra time and effort may still be required to adequately test, understand, and shape system behavior to desired specifications. As in naturally occurring systems, this type of complexity is mainly due to the underlying patterns of collaboration and resource sharing that result due to the formation of sub-networks. Research focusing on studying the properties of discrete dynamical systems such as cellular automata and Random Boolean Networks (RBNs) has yielded many particularly useful insights in this regard. Kinsner describes a dynamical system as one that can be defined mathematically as a set of rules governing the evolution in time t of its *state* in a *phase* [101]. Additionally, dynamical systems can be modeled at discrete time intervals and described in terms of differential equations.

5.1.2 Overview of Elementary Cellular Automata

Cellular automata were originally invented by von Neumann for proving the existence of a self-producing universal computer [107]. Cellular automata have found a vast array of applications, ranging from modeling biological systems, describing populations of mobile organisms, describing the growth of dendritic crystals (such as snowflakes), to studying problems in number theory and their applications to tapestry design. They have also been applied in military contexts for modeling effects-based operations [55]. In addition to being viewed as information processing systems, cellular automata can also be viewed as discrete dynamical systems, or discrete idealizations of partial differential equations [163, 164]. RBN's are a more general case of cellular automata "where the state of each node is not affected necessarily by its neighbors, but potentially by any node in the network" [84]. Wolfram explains the importance of studying these types of systems when he states, "from their analysis, one may, on the one hand, develop specific models for particular systems, and, on the other hand, hope to abstract general principles applicable to a wide variety of complex systems" [163].

The rule states that govern local behavior, which then propagates to a global level, are what distinguish the different classes of behavior expressed by similarly connected elementary cellular automata. An elementary cellular automaton is compromised of any number of individual cells, with each cell either in an on (black) or off (white) state. An elementary cell neighborhood is defined as a middle cell and its adjacent cells on both the left and the right. The state of a middle cell in the next time step is determined by the state of its neighborhood at the previous time step and the particular rule state in use. An example of a rule state can be seen in (b) of Figure28.

An example space-time diagram for the 6 cell lattice shown in (a) is depicted as (c) in Figure 28. Since there are eight possible configurations of states for a three-cell neighborhood and only two possible ways to fill the update table (on or off) for each

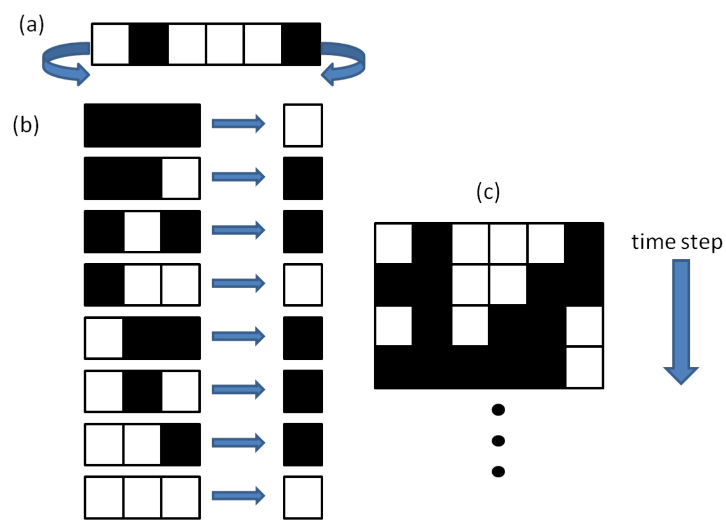


Figure 28: (a) One-dimensional 6 cell elementary cellular automaton whose ends wrap in a circle; (b) Rule table for rule number 110; (c) A space-time diagram, showing three successive configurations. Adapted from [123].

of the eight configurations, there are only $256(2^8)$ possible rules for all elementary cellular automata [123]. For rule number 0, the middle cell is always updated to an off (white) state. Conversely, the middle cells in rule number 256 are always updated to on (black). Wolfram provides numerous illustrations and analyses of evolved space-time diagrams for both elementary and more complex CA's [164]. Wolfram also developed a classification scheme to group the evolved space-time behavior of certain rules from random initial configurations. This behavior ranges from achieving very simple, homogeneous states to stable periodic patterns, chaotic aperiodic behavior, and complex behavior seemingly at the "edge of chaos". A summary of the classification scheme is as follows:

- *Class 1:* Rapid convergence to a uniform, homogeneous state after a finite number of time steps from almost all initial states.
- *Class 2:* Periodic behavior is observed as rapid convergence to a repetitive or stable state.
- *Class 3:* Aperiodic, "chaotic" patterns evolve from almost all initial states.

- *Class 4:* Complex localized structures are observed as locally chaotic patterns interact with stable patterns.

Rule 110, which is a Class 4 rule, has proven to be capable of universal computation. This means that if the initial state is considered as a program and initial data, then the CA is capable of evaluating any computable function [102, 163]. Figure 29 provides an example of evolved space-time diagrams for different Wolfram classes of Elementary CA's. In addition, an example of a naturally occurring Elementary CA pattern is shown.

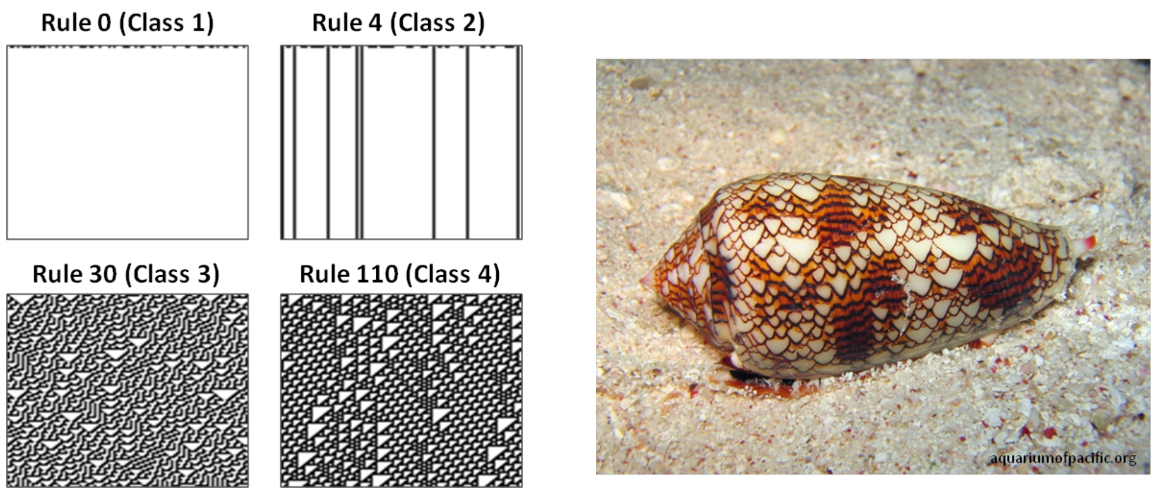


Figure 29: Space-Time Diagrams for Four Wolfram Class Elementary Cellular Automata Rule States & Naturally Occurring Rule 30 Pattern on the Shell of *C. textile*. Cellular Automota Images from [164].

An important parameter for understanding the organizational structure of the Elementary CA rule space is the λ parameter (which should not be confused as an eigenvalue). For cellular automata, the λ parameter is defined as the percentage of all entries in a rule table which map to non-zero states. For cellular automata with 2 states per cell, the rule spaces have a symmetry with respect to $\lambda = 0.5$. This means that rules with $\lambda = x$ are equivalent to rules with $\lambda = 1 - x$ for $0 \leq x \leq 0.5$; the roles of states 0 and 1 are simply reversed [107]. Thus, the λ parameter for the example rule table provided in Figure 28 is $\frac{5}{8}$ or equivalently $\frac{3}{8}$. The λ parameter is analogous, but not directly equivalent to the temperature in statistical physics, or the degree of nonlinearity in dynamical systems [107]. It can also be observed that the particular value of the λ parameter determines the amount of information passed to the middle cell from its left and right neighbors as the update rule is being applied. These two facets of information processing are key features that distinguish the evolution of one lattice from another, and are important features to consider while developing methods to capture the complexities of exchanging resources within a complex system.

Wolfram's classification scheme, though the most widely recognized, has faced some criticism. One of the reasons for this is that not all Elementary CA rules fit neatly within one of the four classes devised by Wolfram, though the scheme holds true in general. This had led researchers in the field to postulate that an additional "phase" parameter is needed to more accurately classify transitions that occur from one class of rules to another within the Elementary CA rule space [32, 102, 104, 107]. This is graphically depicted in Figure 30.

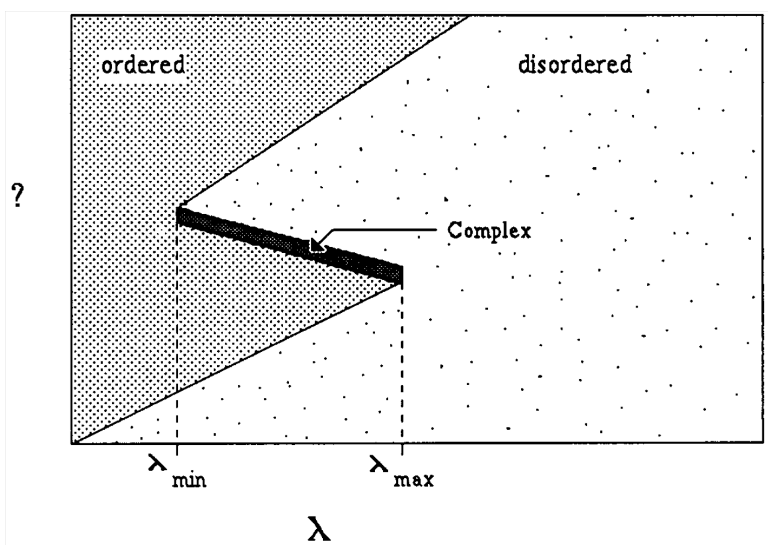


Figure 30: Schematic Diagram of 2D Phase Diagram for CA and its Projection onto the λ Parameter [107].

Ref. [107] describes the dark-shaded region in the middle as a "critical region". Without knowing the value of the second phase transition parameter, experimental results show that both first and second order transitions over a range of λ values. This 2D phase description is analogous to having to know both the temperature *and* pressure of a physical system to understand its transition points. Li *et al.* explain further Ref. [107]:

> For many physical systems exhibiting phase transitions, more than a single parameter is required to accurately reveal the phase-transition structure. For instance, the transition point from a solid to a fluid is not captured precisely by temperature alone, one must also control the pressure. For any specific pressure, there is a unique melting temperature, but if pressure is not being controlled carefully in an experiment, one will observe a range of temperatures at which melting will be observed to occur. This suggests that we will have to find at least one more parameter affecting the dynamics of cellular automata before we can fill out all of the details of the transition or bifurcation structure of cellular automata rule spaces.

As an example, Binder develops an additional independent phase parameter, μ, for Elementary CA's by taking into account the average *sensitivity* of a rule's outcome to small changes in the neighborhood configuration (for example, comparing the change in outcome for configuration (101) vs. (001) for a three cell neighborhood) [32]. Binder also assumes that all neighborhoods are equally probable. μ is used primarily to discriminate between rules with equal λ. Binder observed that it is possible for a rule to have a large sensitivity to changes in neighborhood configuration even though the rule does not allow transmission of information. It has been observed that Class 3 and Class 4 rules are more sensitive than Class 1 and Class 2 rules to changes in site configurations [32]. The notion of sensitivity, especially to initial conditions or configurations is a common theme in the study of chaotic systems [155]. Binder notes that though μ is somewhat crude, it is however directly obtainable from the description of the rule and independent of λ. Thus, it is important to include this aspect of complexity as well. In summary, enumerating the connectivity and the formation of sub-networks that exist between systems provides some insight into the information processing complexity of a system. Yet it proves insufficient as an overall accounting of a system's computational complexity. Elementary cellular automata, for instance, show that homogeneous components, composed in the the exact same structural arrangement can still produce very different behavior.

5.1.3 Complex System Definition

Based on the observations made up to this point, there appears to be a few commonly accepted attributes that tend to be prevalent in describing a system's complexity. These include the number and diversity of system components, the nature of their interconnections and interactions that bind components together, and the difficulty in characterizing the resulting system behaviors [13, 28, 143]. For example, in the field of mechanical engineering design, Summers & Shah refer to these attributes as "size, coupling, and solvability" [152]. Furthermore, there is a recurring theme that arises based on observations of systems that are regarded as highly complex and even chaotic. It has been observed that high levels of complexity/chaos correspond to increased sensitivity to small perturbations. Bak *et al.* note that "dynamical systems with extended spatial degrees of freedom naturally evolve into self-organized critical structures of states which are barely stable [18]. For engineered systems, this means that the current state of the system and the nonlinear relationships that develop over time can cause substantial and unpredictable system reconfigurations [153], making long term planning difficult. For system architects, this makes evolving the system over longer time horizons in a controlled, predictable way more difficult. The end result is that having a good measure of system complexity takes on even greater importance.

In order for an architecture complexity measure to be useful, these relevant properties that contribute to varying levels of complexity must be captured. ***Thus, for purposes of this research, a complex system is viewed as being composed of multiple nodes that are interfaced to share and process resources as part of a cyclic network, and that are able to interact to varying degrees to perform functions and provide a capability.*** To clarify, a node is viewed as an element of either a system or SoS that must interact with other nodes. At the traditional systems level of scope, a node would be considered a subsystem. At the SoS level of scope, a node is what would be usually considered as a system in its own right, such as an F-16 or an aircraft carrier. Also, networks may be either cyclic, meaning they contain closed loops of edges, or acyclic meaning they do not. Newman notes that some networks, such as food webs, are approximately but not perfectly acyclic [126]. This is an important criteria for complex systems, for without cycles present there are no mechanisms for feedback and the emergence of what is commonly viewed as complex behavior. In conclusion, with a clear definition of a complex system in hand, it now becomes possible to evaluate the suitably of different approaches for measuring military SoS architecture complexity.

5.2 Complexity in Relation to Parsimony & Perception

"Everything should be made as simple as possible, but no simpler."

-Albert Einstein

Use of the words simple and complex to describe different systems speak as much to our level of understanding and perceptions about different systems, as well as to the mathematics and techniques used to describe and manage them. From the study of complex systems, we see that the *perception* of what is complex is important to understanding complexity as well. The problem that we encounter, though, is that perception often changes with time and experience. This has prompted many in the field to make clear distinctions between actual and perceived complexity. Summers & Shah provide an apt description when they state [152]:

> Complex systems lie between ordered systems, as in carbon atoms in a diamond, and chaotic systems, as in molecules in a gas [110]. Across this continuum a complexity measure should persist, yet vanish at the extremes. This is a naïve view in that the molecules in a gas may be ordered and acting according to specific laws of physics as yet not discovered. Thus chaos, in these terms, may simply be a description of a complex system that is as yet not understood.

In similar fashion, Crawley makes a distinction between essential, perceived, and actual complexity. In Crawley's view, "it is necessary to keep perceived complexity below the limit of understanding (comprehension) and the actual complexity close to the essential complexity. The actual complexity is never smaller than the essential complexity" [100]. Likewise, Suh postulates that there is an imaginary component to complexity in addition to an observable, real component. According to Suh, "This imaginary complexity is a complexity that is not real, but exists because of our lack of understanding about the system design, system architecture, and/or system behavior" [151]. The conclusion that can be drawn is that one's current level of knowledge, expertise, and technological know-how is a primary factor in shaping one's perception of what is simple vs. what is complex. This is in agreement with Summers & Shah's research in the field of mechanical engineering design, where design is viewed as "a cyclic process that includes synthesis, analysis, and decision-making stages to migrate from design problem to design product" [152]. Furthermore, the design problem is a statement of the requirements, needs, functions, or objectives while the design process includes the steps that are undertaken to find satisfactory solutions to the stated problem. Designer experience as well as established rules, procedures, and domain knowledge are all valid resources that may be utilized during the execution of the process. Lastly, the design product is "a representation of the envisioned physical solution to the problem through the realization of the dependent design variables such that the design relations are satisfied" [152].

A simple example using the automobiles depicted in Figures 31 and 32 further illustrates the relationship between design problem, process, and product in terms of the effect on actual vs. perceived complexity. Henry Ford, for instance, developed the Model T automobile during the 1900's. If, during the 1900's, he could have been presented with an automobile from a later time such as the 1960's or 1970's, he would most likely view such an automobile as a step forward in manufacturing and design complexity. In contrast, today's automakers most likely consider that same automobile from the 1960's and 1970's as relatively much simpler in comparison to today's vehicles, especially when modern automobiles incorporate into their design the use of integrated electronics and computerized sensors, more advanced materials such as plastics, novel propulsion plants such as hybrid gasoline/electric engines, and so forth. The reason for this shift in perceived complexity is that both the design problems and processes encountered by automobile manufacturers have changed dramatically from the time of the Model T to today. In terms of the design problem, the automobiles of today must meet different requirements and provide different functionality in comparison to past automobiles. Increased fuel efficiency standards, crash-safety ratings, and the ability to interoperate with portable communications and electronic devices are but a few. Likewise, the design process itself is influenced by the education and training of the industry workforce, affecting the levels of proficiency and efficiency that can be brought to bear in the allocation of resources.

These considerations provide an important insight that will prove useful later when attempting to monetize complexity. *That is, complexity can be discounted at least in part based on the perception of the difficulties and risks inherent in migrating*

Figure 31: 1910 Ford Model T and a 1970 Ford Torina Cobra.

Figure 32: Exterior and Interior Views of the 2010 Ford Fusion Hybrid.

a system through the different phases of design, beginning with the conceptual design phase and ending in a physically realizable system. From this example it is also important to consider that even though the perception of a system's complexity may change over time, there is one aspect of a system's complexity that can be considered an absolute property, meaning it is independent of the observer [100]. From this point of view, as a design product, the 1960-1970's era automobile did not change in terms of the functionality it provides, the number and types of interfaces used to interconnect its many sub-components, or any other similarly objective system parameters that speak directly to the makeup of that particular vehicle. This brings up an important assumption that should be discussed when speaking in terms of architectures. An architecture is assumed to be defined in terms of the set of functions that compromise the required end-to-end capability. With this being the case, it is possible for the complexity of an architecture to remain the same even though the complexity of a system within the architecture changes. The caveat is that this is dependent upon the required functionality that defines the architecture remaining the same. For example, a bicycle and an automobile might be limited to providing the same function within an architecture, with that function being to transport one person one mile. Intuitively, an automobile is thought to be more complex than a bicycle. However, since either system can meet the required functionality as it is specified, from this strict perspective trading one for the other does not change the overall complexity of the architecture. Here, other considerations such as cost, maintainability, and flexibility may come into play.

Hence, when faced with choosing between a highly complex product design vs. a much simpler one as a solution to a design problem, a decision maker that only requires basic functionality may very well decide on a parsimonious solution to the decision problem. The principle of parsimony, which is the preference for the simple over the complicated or complex whenever possible, is a guideline that is often adhered to in science and engineering. In science there is a preference for the simplest plausible explanation of an observation. Meanwhile, engineering designs that meet functional requirements in the least complex manner are preferred [33, 132]. In essence, complexity should be limited to that which meets the necessary system requirements for improved functionality, reliability, etc. This task becomes more complicated, however, when other pertinent system attributes such as affordability and flexibility must also be considered. Here, the architecture again plays an important role in helping to define just how adaptable and flexible the system will prove to be [52, 69]. This is all the more reason to ensure that system architects are equipped with a quantitative, objective measurement of architecture complexity when comparing alternatives. Besides clarifying the intuitive sense of complexity and making the design problem more transparent, it can also aid decision making by helping to better inform tradeoff analyses, while also ensuring the strategic allocation of resources occurs in order to effectively manage complexity, reduce risk, and prevent negative acquisition outcomes.

5.3 Approaches to Measuring Complexity

Over time, notable descriptions of complexity have ranged from organized vs. disorganized complexity pioneered by Weaver in 1948 [162] to the algorithmic complexity of data structures developed in the 1960's by Kolmogorov, Solomonoff, and Chaitin [100] to Suh's uncertainty-based characterization of complexity [151]. This has lead many researchers in the field of Complexity Science, as well as those in disciplines that frequently encounter and analyze systems considered to be complex, to formulate a wide variety of complexity measures. Likewise, many researchers provide well-researched listings and critiques of many of the most popular complexity measures [36, 76, 123]. Lloyd states that intuitively, an ideal measure of complexity would be universal and applicable to any dynamical system—whether living, nonliving, or artificial [110]. Others note that many of the existing complexity measures developed by complexity scientists tend to be very domain specific or too theoretically abstract to usefully apply to real world systems [13, 75, 123]. Perhaps the overarching reason for this is that the diversity that exists among both natural and engineered systems makes it difficult at best to define an absolute measure of complexity that is applicable to any and all systems [101, 123]. As Kinsner states, "Complexity appears to be context sensitive, and cannot be defined universally, once and for all" [101]. Rather than trying to adapt or define a single, universal measure of complexity when evaluating military SoS architectures, the focus will be to capture the specific traits that have general consensus as primary contributors to what is commonly viewed as complex behavior.

Though there is a large and growing number of ways to measure complexity, both Kinsner and Lloyd demonstrate that many of these represent variations on a few underlying themes [101, 109]. This makes it possible to classify many existing measures into a manageable number of categories. A particularly useful classification scheme devised by Lloyd points out that in general, researchers frequently ask the following three questions to quantify the complexity of an object under study [109]:

1. How hard is it to describe?
2. How hard is it to create?
3. What is its degree of organization?

Many of the measures that seek to address the first question usually do so in terms of measuring the amount of randomness exhibited by a system. This has led to new theories in describing the link between information and entropy [147]. Thus, these measures tend to focus on patterns that signify the amount of disorder or irregularities exhibited by a system. Within this class of measures, an alternative approach to measuring entropy advocates that the complexity of an object is also related to the size of the shortest computer program that could generate a complete description of the object. This, of course, requires that the object in question can be represented in the form of messages such as a string of data [123]. One example of such a string could be a binary sequence of zeros and ones such as 0101010101 or 1100100010. Another common example is the representation of a DNA sequence where the letters A, C, T, & G are used to represent nucleotide bases. There is general agreement that these measures of structural complexity should have the property that there is low measured complexity for both very ordered, regularly repeating patterns and for random patterns as well [101], since they each convey very little information. Gell-Mann accomplishes this, for instance, by only specifying the information content of the string's *regular* patterns. In this way, a purely random string has no regularities and thus no effective complexity [82]. In between are patterns of various levels of complexity that possess both patterns of regularity and randomness. While complexity is related to the amount of information needed to describe a phenomena, measuring complexity in this way for engineered systems poses certain difficulties. Chief among them is that to do so requires a consistent means of representing the physical system itself as messages. Even when this is accomplished, the definition of patterns of regularities and irregularities can be very subjective and difficult to define [123].

Complexity measures geared toward answering the second question attempt to assess the complexity of an object by measuring how difficult the object is to construct. Within this class of measures there are varied metrics ranging from time, energy, dollars, and computational resources. Even the "logical depth" of a system can be considered, which is a measure of how difficult the object is to construct or reproduce using the most plausible method of creation [29, 123]. This is at odds with assessing the complexity of an acquisition during a capabilities-based conceptual design effort, even if the required functionality is known. To accurately estimate complexity in this way requires highly detailed information which is usually not available at this stage of design, especially in the case of newly developing systems and interfaces. For example, the amount of time, energy, and dollars used to either construct a military SoS or integrate new components into an existing SoS is heavily dependent on many dynamic factors. These include the materials, technologies, manufacturing methods, logistics, and T&E processes to name a few. If this information could be predicted well in advance, then selecting the best architecture becomes a simple cost-effectiveness comparison. Ultimately, the system architect is attempting to infer this aspect of complexity ahead of time from the limited knowledge that is possessed at the conceptual design phase.

Braha *et al.* state that "An essential tool for understanding complex systems is to study the system's organization, which is often relatively simple. Understanding the organization of the system can also lead to a better understanding of the system's behavior" [36]. Therefore, addressing the third aspect of complexity is more directly applicable to system architecting. Lloyd points out that measuring the degree of organization can be separated into two fundamentally important aspects. The first is the difficulty of describing the organizational structure of the object under study.

The second focuses on the amount of information shared between the parts of a system as a result of the organizing structure [109]. This is directly applicable to SoS, where the added value of SoS capabilities are "derived from the interaction among components rather than from contributions of the individual components" [68]. Consequently, the primary goal of the system architect is to determine the best arrangement of system functionality and resource sharing. Once achieved, this makes it easier to conduct testing, verification, and maintenance of the system while ensuring robust functionality in dynamic acquisition and operating environments. While the specific measures that are catalogued under this third category still tend to be information-theoretic or entropy based, the general premise proves useful and provides an important conceptual underpinning on which to develop a framework applicable to military SoS. Measuring the complexities of the combined interactions and resource sharing between systems will provide the necessary understanding of the behavior and performance of the SoS. This is critical to successful SoSSE [69].

5.4 Measurement Criteria

McCabe & Butler, while developing measurements that quantify the architectural complexity of different software designs, present additional criteria that can be used to judge the applicability of any proposed measure for quantifying the architectural complexity of a complex system [119]:

1. *The metric intuitively correlates with the difficulty of comprehending a design i.e.*, when we view large complicated designs, the metric should yield a high number. Designs we intuitively deem as simple should have a relatively low number.

2. *The metric is objective and mathematically rigorous.* The same design viewed at two different times or by two people should yield the same complexity.

3. *The metric should be related to the effort to integrate the design.* The proposed metric should correlate directly with the cost and effort experienced in the integration phase.

4. *The metric should help generate an integration test plan early in the life cycle.* If the metric can be computed in the design phase, a set of tests that are derived from the design can traverse the architecture in a rigorous way.

5. *The metric and associated process should be automatable.*

These requirements may seem obvious at first glance, however, ensuring that these criteria are adequately addressed are essential to confirming the utility of the measure that will be later applied.

5.5 Existing System Complexity Measures

During the course of this research some notable methods of measuring system complexity were encountered. In particular, Summers & Shah provide a valuable survey of different approaches for measuring complexity in engineering design, evaluating the applicability of each in terms of measuring the complexity of design problems, processes, or products [152]. What follows in the ensuing sections is a brief discussion of different measures deemed potentially the most relevant in helping to measure the complexity of military SoS architectures. Each of the existing system complexity measures are evaluated according to the criteria put forth by McCabe & Butler. They are also evaluated against their potential suitably for use in conceptual design.

5.5.1 Abstraction Based Complexity Management

An abstraction based complexity measure developed by Zeidner *et al.* for managing complexity in aerospace systems [28, 166, 167] provides many useful features & principles for measuring system complexity. This method of measuring system complexity is focused on using parameters that are readily available to the designer to assess the complexity of a vehicle resulting from the required interactions of its subsystems. The abstraction based complexity measure possesses a distinct advantage over the preliminary method of measuring complexity used by DARPA's META program, for instance, where META uses part count and number of source lines of code. This is because the abstraction based complexity measure seeks to avoid the oversimplification of complexity that occurs when only considering the part count of the number of distinct system components and interconnects that exist within the system. The rationale for this reasoning is that "it is of course possible to design a system that consists of many interconnects, and yet is not complex" [28]. As a result, the abstraction based complexity method assigns complexity weighting factors to each node (subsystem/component) and interconnect during the summation of each. The complete formulation is presented as the following:

$$C\left(n, A\right) = \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{4} \beta_k \alpha_{ijk} + \gamma \left[\frac{\log n}{\log 7}\right] E(A) \tag{5}$$

where:

- α_i = component complexity

- β_k = interconnection complexity

- α_{ijk} = an interconnected component

- n = level of abstraction

- $E(A)$ = graph energy

The first term in Equation (5) is a weighted sum that not only represents the number of system components under consideration, but also the inherent complexity associated with each individual component. Because of this, system architects can gain a better sense of the important tradeoff that occurs when component reduction is achieved at the expense of fewer, more complex components [28, 166, 167]. The second term captures the number of interconnects/interfaces present as well as the relative complexity of each interconnect. Inclusion of this second term provides a distinct advantage over Technomics' method, which does not make a distinction between the contribution to complexity of different types of interfaces. Without this distinction, it is possible that an architecture with very few, but very complex interfaces to support resource sharing could be rated as being relatively less complex than an architecture with very many, but very simple interfaces [167].

The final term in Equation (5) contains two distinct quantities that are multiplied together (It is not clear what γ represents beyond being an additional weighting factor). The level of abstraction currently under consideration, denoted as "n", is based on the assumption that from the highest to lowest level of abstraction there are seven layers. This appears to be modeled after the abstraction layers developed for the Open Systems Interconnection (OSI) standard developed by the International Organization for Standardization (ISO). The seven abstraction layers in the OSI Reference Model define the movement of data over a network. For example, the first layer in the OSI model is the physical layer which provides mechanical, electrical, functional, and procedural characteristics to establish, maintain, and release physical connections between data link entities. In contrast, the seventh and highest layer is the application layer where direct service is provided through interaction with an operating system or application [169]. Layering provides a structuring technique so that the network of open systems can be viewed as logically composed of a succession of network layers, each wrapping the lower layers and isolating them from the higher layers [169]. The value of n used in Equation (5) is based on a heuristic decomposition of typical aerospace vehicles, requiring refinement for other systems under consideration. The graph energy term, $E(A)$, captures the density of connections in the system with respect to the total number of possible interconnections [19]. Graph theory can be used to render a system in mathematically abstract terms, where collections of vertices are connected or linked together by edges. An adjacency matrix can be formulated to also describe the patterns of connections between vertices [42]. Figure 33 provides a visual depiction of a generic system made up of nodes A, B, and C and the corresponding adjacency matrix. Two cases are shown. Case (a) is a simple *undirected* graph. Case (b) is a *directed* graph (or *digraph*) with a loop. Mixed graphs include a combination of directed and undirected edges. Also, the adjacency matrix for a simple, undirected graph is the same as that of a bi-directional digraph with no loops.

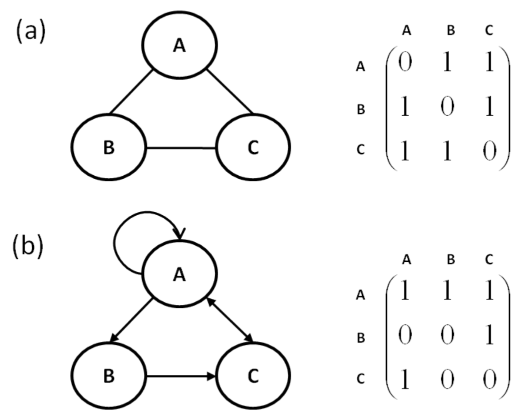


Figure 33: Example Graphs & Adjacency Matrices.

Cares remarks on the usefulness of using eigenvalues derived from a graph's associated adjacency matrix for analysis [42]:

> A very rich and formal field of mathematics exists to perform these operations. One of the most useful operations is the calculation of eigenvalues. An eigenvalue, usually denoted by the Greek symbol, λ, is a measure of the value of the network and is derived from the adjacency matrix.

The energy of a graph is defined to be the sum of the absolute values of the eigenvalues of the adjacency matrix [19]. It is intended to be a measure of the dynamic instabilities that can increase as both systems and time scales become more interconnected [28]. This same principal is also applied in Technomics' formulation of measuring complexity. Together, these three factors (weighted node complexity, weighted interconnect complexity, and abstraction-weighted connection density) are summed together to present an overall view of system complexity. An important technical note to discuss is that when values are assigned to the links and nodes of a graph, a system with its own logic is created. This system is more properly defined as a *network*.

In evaluating the abstraction based complexity measure as a whole, the summation of these submeasures proves problematic in a sense. Theoretically, one could define systems with either zero node complexity or zero interconnections between nodes but that are still rated as complex. For example, a system whose behavior is easy to predict is one in which the nodes are either lacking in functionality or lacking the necessary interfaces that allow the passing of information or other resources between nodes to carry out the intended functionality. Based on this description, it would be difficult to justify this type of system as being a complex system. Yet the summation of the submeasures could mathematically lead to such a system being deemed more complex than one with relatively fewer functioning nodes and interconnects. This violates the first criterion proposed by McCabe & Butler. In addition, since a linear combination is used one must pay careful attention to the relative weightings of each of the three terms in Equation (5) in relation to each other. Failure to do so could easily result in one term incorrectly dominating the entire complexity calculation.

A final observation is that this measure is more characteristic of a descriptive approach to measuring complexity. As such, it may be more appropriately used when there are well-defined subsystems. This is mainly due to the fact that the measure was developed with the goal of researching existing drivers of complexity in aerospace systems. As a result, this particular approach assumes that a certain level of technical detail about the constituent subsystems is available when making comparisons.

Consequently, adapting this measurement to capability-based conceptual design may prove difficult in certain cases. This is especially true as the number of novel and disparate systems that are candidates for incorporation into the SoS architecture increases.

5.5.2 Object-Process Model Based Complexity Measures

Kinnunen describes various complexity measures for system architecture models [100]. These measures rely on an Object-Process Methodology (OPM) modeling language designed to model any system as a collection of objects, processes, and states. Dori describes this relationship as "Objects exist, and processes transform the objects by generating, consuming, or affecting them. States are used to describe objects, and are not stand-alone things" [73]. By using the same coding language to represent system architecture models, the relative complexity of each model can be compared by assuming that the more difficult the object is to specify or describe, the more complex the object is. This necessitates deciding upon the level of abstraction of the coding language, as lower levels of abstraction incorporate more details into the analysis and result in higher model complexities. Separate system architecture complexity measures are then applied at a particular level of abstraction. They are based both on existing complexity measures and on the system architecture models. They include the following:

1. *Number of distinct types of things*: The greater the number of types of things (objects, processes, and state) a model has, the longer the program is needed to produce the model.

2. *Sum of number of things of each distinct type*: The more instances there are of a certain type of thing, the longer the program needed to produce the model. The length of the program does not depend linearly on the number of instances. If there is one instance, that instance is counted once. If there are several of the

same instance, that instance is counted twice. In effect, the program to produce a model with many instances is twice longer than a program to produce a model with only one instance.

3. *Number of processes affecting an object*: The more processes that affect a given object, the longer the program needed to produce the model. The minimum, maximum, and average values are calculated. If the minimum is zero, the model has a dangling object. If the maximum is much higher than the average, the model may have an uneven distribution of complexity.

4. *Number of objects being affected by a process*: The more objects a process affects, the longer the program needed to produce the model. This is very similar to the previous elementary measure.

5. *Number of operands per process*: The more operands a process has, the longer the program needed to produce the model. Again, minimum, maximum, and average values are calculated, since high average and high maximum values may indicate an unevenly or highly complex system.

6. *Number of interfaces weighted by Interface Complexity Multipliers (ICMs)*: The more interfaces the model has, the longer the program needed to produce the model. ICMs are used to compensate for hidden information due to abstraction.

Regarding these measures, Kinnunen states, "With any real life systems and their models, the results are very likely to be contradictory. A model is more complex than another according to one measure, but less complex according to another measure. For this reason, lumping measures together with any formula is not going to be meaningful in all situations" [100]. Prioritization of the measures is left to the system architect. Though this measure as a whole meets the McCabe & Butler criteria, it is not immediately clear how simply providing a more detailed summation of objects, processes, and states provides system architects with additional insight into the more dynamic aspects of complexity. These dynamic aspects are a result of variations in the sequence of functional execution and from changes in the patterns of resource sharing and processing. This information may exist in the OPM models themselves, but are not quantified in a way from which clear comparisons can be made. The analysis is made more complicated when comparisons across multiple level of abstractions must be made, as is done by Kinnunen [100]. Finally, Kinnunen explains that further use of the methodology requires extending the OPM modeling language, since measuring the complexity of the system architecture models is quite tedious.

The OPM modeling methodology can be characterized as a measurement that describes the complexity of engineered systems in terms of how hard it is to describe or create an object. The methodology does present some useful aspects, however, one being that this measurement approach lends itself easily to automation. Once the structure of the model is known, the structural relationships between objects, processes, and states can be represented by a matrix. Use of the matrix format opens up the space of graph-theoretic and linear algebra techniques as comparative analysis tools. This approach also emphasizes the importance of taking into account object states. In order to fully understand complex system behavior an understanding of the particular states the system can inhabit and the factors that directly affect the occurrence of state transitions is necessary.

5.5.3 Suh's Axiomatic Design

Suh defines complexity as "A measure of the uncertainty in understanding what it is we want to know or in achieving a functional requirement (FR)", where a FR is defined as "a minimum set of independent requirements that completely characterize the functional needs of the product" [151]. Both COSYMO and CoBRA adhere to the same principle, as they each attempt to measure complexity by capturing the uncertainty that arises due to the number and level of requirements placed on a design. It was shown previously how the number of system requirements and the level of requirements understanding are similarly included as size and cost drivers in the COSYMO parametric cost model. Likewise, CoBRA presents a method of quantifying complexity based on a historical perspective of the impact of system requirements on satellite design. As Braha *et al.* make note of, though, "Suh's conception of complexity in design as having to do with uncertainty, while potentially useful, is qualitatively different from the kind of complexity we appear to be confronting in design, which involves interactions and relationships, not just uncertainty" [36]. A particularly useful concept that arises from Suh's formulation, however, is the minimization of complexity by reducing the coupling between design parameters (DPs). In this context an uncoupled design is one in which the FRs are independent in relation to one another as they map to physical DPs. This mapping is achieved through a *design matrix*, $[A]$, that relates FRs to DPs and characterizes the product design. When the set of FRs and DPs are described in vector form, the following equation results:

$$\{FR\} = [A]\{DP\} \tag{6}$$

Suh's Independence Axiom states that when there are two or more FRs, the design solution must be such that each of the FRs can be satisfied without affecting any of the other FRs. This means choosing a set of DPs that can satisfy the FRs and maintain independence. Depending upon the relative numbers of DPs and FRs, a design can be classified as coupled, redundant, or ideal. When the number of DPs is less than the number of FRs, the result is always a coupled design. Conversely, a redundant design is always the case where the number of DPs is greater than the number of FRs. Finally, an ideal design is an *uncoupled* design where the number of DPs is equal to the number of FRs *and* the FRs are kept independent of each other. For an uncoupled design the design matrix is diagonal:

$$\begin{Bmatrix} FR_1 \\ FR_2 \end{Bmatrix} = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \end{Bmatrix} \tag{7}$$

In this case, the Independence Axiom is satisfied and will lead to a less complex design. If the number of FRs matches the number of DPs, but the design matrix is triagonal:

$$\begin{Bmatrix} FR_1 \\ FR_2 \end{Bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \end{Bmatrix} \tag{8}$$

The independence of the FRs can be guaranteed if and only if the DPs are determined in a proper sequence. Suh refers to this type of design as *decoupled.* It follows that as the relationships between FRs becomes more coupled so does complexity. Suh makes note that the design effort may produce several designs that are acceptable in terms of the Independence Axiom. Therefore, in addition to the Independence Axiom, Suh postulates an Information Axiom. The Information Axiom states that the design with the smallest information content is the best design, since it requires the least amount of information to achieve the design goals. In this way, a physically large system is not necessarily complex if the information content is low, but a small system can be complex if the information content is high. The information content for a given FR is measured by specifying a probability of achieving the design goals.

Similar to Perrow, Suh specifies time-dependent aspects to complexity in his axiomatic formulation. Since systems must operate within an environmental context, perceived changes within the operational environment are often communicated as changes to system requirements. Often this results in the system being forced to adapt and evolve, or risk becoming obsolete. While the difficulty in achieving a set number of functional requirements may be debatable as a sole measure of complexity, undoubtedly an understanding of how requirements relate to each other must be taken into account when attempting to architect complex systems and guide their development.

5.6 Architecture Complexity Sub-Measures

The survey of existing system complexity measures yielded the potential benefits and drawbacks that are characteristic of each. This knowledge can be leveraged to develop an approach for identifying and measuring the chief system properties that can be readily observed and that describe a military SoS architecture in terms congruent with the previously developed complex system definition. For example, a fundamental feature of military SoS architectures is that they are comprised of physically distinct systems, each performing certain tasks, that combine together to provide a capability set. A capability is defined as "the ability to achieve a desired effect under specified standards and conditions through combinations of ways and means to perform a set of tasks" [46]. Though system level tasks can be further aggregated into higher level activities, a primary way of distinguishing one military SoS architecture from another is through the mapping of system level tasks to the physically distinct systems that make up the SoS. Since the behavior of a SoS is expressed in terms of functionality, or the ability to meet assigned tasking, it then makes sense to measure the complexity of the SoS in terms of functionality as well. It is important to note that higher level activities must be decomposed to an appropriate level of scope so that at least one of the individual component systems under consideration is capable of completing the task. As previously discussed, information processing/computation is another important aspect of complexity, especially when it involves the parallel or decentralized computation that can occur from multitudes of networked components [123, 163]. Moreover, the primary reason for applying network-centric principles to military SoS is to foster the ability to "share, access, and protect information to a degree that it can establish and maintain an information advantage over an adversary" [11]. Within these networks, it is the aggregation of localized interactions that is the dominant, observable feature that gives rise to complex behavior. Because each system function within the SoS architecture can be associated with a set of localized information and service exchanges, it is logical to extend this concept to resource flows in general. Subsequently, four aspects that contribute to military SoS architecture complexity and their accompanying sub-measures are the following:

1. *System Physical & Functional Boundaries*: This aspect captures the organization and distribution of functionality among systems within the SoS. The accompanying sub-measure will be referred to as the Functional Distribution Complexity, or **FDC**.

2. *Functional Process/Task Sequencing*: This aspect describes the patterns of system-to-system interactions dictated by the existing functionality. The accompanying sub-measure will be referred to as the Functional Processing Complexity, or **FPC**.

3. *Resource State Characteristics*: This aspect characterizes the properties of and effects from the particular resources being exchanged to ensure functionality is met. The accompanying sub-measure will be referred to as the Resource State Complexity, or **RSC**.

4. *System-to-System Interfaces*: This aspect delineates the patterns of collaboration and resource sharing that enable cyclic behavior to occur. The accompanying sub-measure will be referred to as the Resource Processing Complexity, or **RPC**.

Defining context appropriate, and thus relevant, complexity sub-measures for each of these aspects will provide the basis for developing a suitable measure of SoS architectural complexity. Hence, both contextual nuances specific to an architecture and the nuances exhibited by different aspects of the notion of complexity itself can be captured using appropriate metrics, whether they be pre-existing or newly defined. The following sections will explore each of the four aforementioned aspects in further detail in order to demonstrate the options available for developing valid military SoS architecture complexity sub-measures.

5.6.1 Functional Distribution Complexity

SoS architectures can vary in the distribution of functionality among component systems. Consequently, one architecture may be composed of a number of relatively simple systems, each with very limited functionality while another architecture may rely on very few systems, each capable of fulfilling many different tasks. Also, systems may have overlapping functionality. In the case of the first architectural approach, where functionality is widely and evenly distributed with little or no overlap or redundancies, fulfilling the performance requirements becomes much easier. The reason for this is that there is a greater degree of independence in pursuing an optimal solution. Alternatively, combining multiple functions into a single system raises the possibility of competing SoS behavioral effects and system design compromises. Perrow provides an insightful example to describe this condition [136]:

> But what if parts, or units, or subsystems (that is, components) serve multiple functions? For example, a heater might both heat the gas in tank A and also be used as a heat exchanger to absorb excess heat from a chemical reactor. If the heater fails, tank A will be too cool for the recombination of gas molecules expected, and at the same time, the chemical reactor will overheat as the excess heat fails to be absorbed. This is a

> good design for the heater, because it saves energy. But the interactions are no longer linear. The heater has what engineers call a 'common-mode' function—it services two other components, and if it fails, both of those 'modes' (heating the tank, cooling the reactor) fail. This begins to get more complex.

Perrow also notes that "Ironically, in many cases, the complexity is added to reduce common-mode failures. The addition of redundant components has been the main line of defense..." [136]. As a result, FDC seeks to capture the complexity that results from the tradeoff in component reduction when it is achieved at the expense of fewer, more complex systems [28, 166]. Here, complexity is measured by the distribution of functionality among the physically & functionally distinct node types. Figure 34 further illustrates what is meant by functionally & physically distinct nodes.

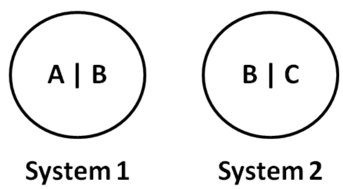


Figure 34: Example of Functionally & Physically Distinct Systems with Overlapping Functionality.

In Figure 34 each circle represents a physically separate system while the letters A, B, and C represent different functions that must be completed. Though Systems 1 & 2 have overlapping functionality they are still functionally distinct. This is because System 1 differs from System 2 in the entire set of functions it can fulfill. An additional system that exactly duplicates the functionality of either System 1 or System 2 would not meet this criteria and would therefore not be considered in the FDC calculation. Within an architecture there may be multiple instances of a system type, and in terms of functionality, functionally equivalent system types are considered equal within a functional architecture.

Now that individual functional nodes are properly described and identified, the next step is to determine a way to quantify FDC, or the contribution to overall architecture complexity due to the distribution of functionality across nodes. In order to prove useful, the FDC sub-measure should also capture the added complexities that arise due to functional integration. More specifically, the FDC sub-measure must follow the premise that complexity increases when combining multiple functions into a single node, since at a minimum those functions will be utilizing common resources such as physical space, power, or computer memory. Lastly, the level of the functional requirements or the specified tolerances a system must meet in providing the desired functionality plays an important factor as well. It is assumed that an architecture that specifies a high level of performance in meeting requirements and that has tight tolerances will be more complex to create, maintain, and operate [151]. Requirements can be specified at the level of each function and at the higher SoS-level. Following these assumptions, a general form of the FDC can be expressed as follows:

$$FDC = \frac{\sum_{i=1}^{T} F_i^{\rho}}{T} \tag{9}$$

where:

- T = number of functionally & physically distinct nodes

- F_i = number of functions performed by the i^{th} node

- ρ = functional integration exponent

Figure 35 provides an example of alternative functional allocations for a set of generic functions labeled A-E. For these examples, the following equations demonstrate how Equation (9) captures the tradeoff in component reduction when it is achieved at the expense of fewer, more complex systems.

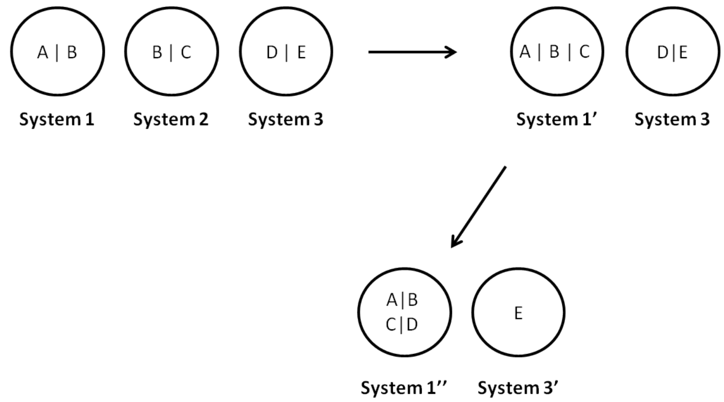


Figure 35: Example Tradeoff In Component Reduction With Increasingly Complex Systems.

$$\left(\frac{\sum_{i=1}^{T} F_i^{\rho}}{T}\right)_{Alt1} = \left(\frac{2^2 + 2^2 + 2^2}{3}\right) = 4.0 \tag{10}$$

$$\left(\frac{\sum_{i=1}^{T} F_i^{\rho}}{T}\right)_{Alt2} = \left(\frac{3^2 + 2^2}{2}\right) = 6.5 \tag{11}$$

$$\left(\frac{\sum_{i=1}^{T} F_i^{\rho}}{T}\right)_{Alt3} = \left(\frac{4^2 + 1^2}{2}\right) = 8.5 \tag{12}$$

For ease of comparison, a value of $\rho = 2$ is arbitrarily chosen for each of the alternative functional allocations in Equations (10) – (12). For the FDC calculation, ρ represents the relative difficulty of integrating specific groups of functions into a single physically & functionally distinct node. The value of ρ should be equal to 1 for designs where the estimated difficulty of functional integration is low and increase as functional integration is estimated to create additional nonlinearities, couplings, and complex interactions. The following are examples of the different approaches that can be used for determining ρ. In the future, new approaches or techniques may be developed (possibly including new combinations of existing approaches), expanding the options available to system architects.

1. Direct Function-to-Function Comparisons

2. Uncertainty-based Analysis

3. Graph-based Variable Dependency Analysis

4. Historical Database Estimating

5. Matrix-based Correlation Analysis

5.6.1.1 Direct Function-to-Function Comparisons

The first, and perhaps most obvious method of determining ρ, is to perform direct function-to-function comparisons to analyze the impact on SoS-level behavior resulting from nodes with increased multi-functionality. This is similar to Perrow's tank and heater example given earlier, and results in a separate ρ for each node. This can be expressed as ρ_i, or the functional integration exponent for each i^{th} node. While this approach is straightforward, it should be reserved for SoS architectures where the functional decomposition yields a low number of system-level tasks or functions to be executed. To conduct a thorough analysis, pairwise comparisons of each function-to-function relationship for a group of functions must be made. The nature of this approach creates the potential to quickly make such an endeavor a tedious effort, especially as the number of functions increases. To illustrate, the number of possible relationships based on function pairings is nonlinear itself and best expressed by $p = n^2 - n$, where n is equal to the number of functions. Consider, for example, a system consisting of only 10 functions. Using the aforementioned formulation, $p = 90$. Simply doubling the number of functions to 20 results in $p = 380$. It is important to keep in mind that military SoS architectures can easily encompass hundreds of functions depending upon the desired mix of capabilities. The problem becomes even more difficult if the exact strength of each function-to-function relationship must be adequately assessed, and even more so if the differentiation between the effects of primary and secondary interactions must be included in the analysis.

5.6.1.2 Uncertainty-Based Analysis

As explained in Section 5.5.3, Suh defines system complexity in terms of the measured uncertainty in achieving FRs. Suh's Information Axiom can be adapted for use in determining ρ, assuming that there is a correlation between information content and the effort expended for successful functional integration. As more complicated procedures, policies, resource allocation, and design compromises become necessary to ensure successful integration, this translates directly to greater information content. (Here, instead of the term FR, the terms required functionality or functional requirement will be used to specify requirements for system-level functions and differentiate between SoS-level performance requirements, and to avoid confusion with Suh's precise meaning and use of the term FR). Using Suh's formulation, the general case of determining the information content I_i for F_i number of functions performed by the i^{th} node is the following:

$$I_i = -\sum_{m=1}^{F_i} \log_b P_m \tag{13}$$

where P_m is the probability of satisfying the functional requirement. P_m is based on which system will be providing the functionality and the specified levels and tolerances imposed on the design. Common choices of the base b for the logarithmic function are usually 2 (with the unit of bits), 10, or the natural logarithm (with the units of nats), depending upon the system architect's preference. The logarithmic function is commonly used in information-based analyses due to its many unique features. In this instance, the logarithmic function is chosen so that the information content will be additive when there are many functional requirements that must be satisfied simultaneously. When all probabilities are equal to 1.0, the information content is zero and I_i is zero. Conversely, the information content is infinite when one or more probabilities are equal to zero. This gives an infinite value for I_i as well. It should be noted that Equation (13) is a simplified form that assumes independence between each P_m. To assume otherwise might lead to overestimating the degree of difficulty due to functional integration, since it is already assumed that multiple functions within a node may cause nonlinear interactions, based on the summation of the values of P_m within a particular node. With that being said, the following equation can be used to determine ρ:

$$\rho = I_i + 1 \tag{14}$$

In terms of Perrow's example, if the temperature bands that must be maintained in the tank and chemical reactor are very wide and well within the functionality capacity of the heater, then the probabilities of successfully meeting the heating and cooling requirements will be high. In this case, ρ will approach a low value. On the other hand, if tight tolerances must be maintained, then the information content will be high as a result of low probabilities. The resulting large values for ρ will mean added integration complexity if these functions are made part of the same system. The main concern with using this method of determining ρ is that it makes preformed assumptions on what types of systems will provide certain functions in order to calculate the probabilities. System architects must be careful not to eliminate certain portions of the design space prematurely when using this method, especially when performing capabilities-based design.

5.6.1.3 Graph-based Variable Dependency Analysis

From the perspective of design, each function can be mapped to different design variables. These include independent variables, dependent variables, and design relationships, for example. Independent variables are variables whose values are controlled by the designer. Meanwhile, design relationships are constraints that dictate the association between other design variables. Lastly, dependent variables are those whose values are not directly under the control of the designer, but that are derived from the independent variables, other dependent variables, and design relationships. Based on these mappings, the connections between variables at multiple levels can be represented in a graph-based format. An example is provided in Figure 36 where entities such as design variables are represented by circles and the relationships are represented by squares.

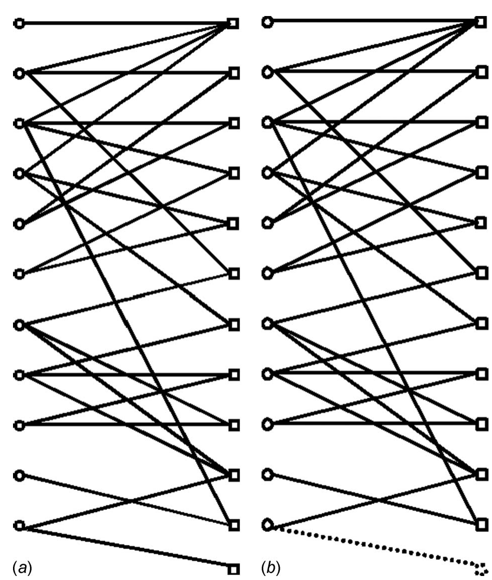


Figure 36: Example Entity-Relationship Graph: (a) Initial graph and (b) Removed unary constraint [152].

Summers & Shah provide an approach to measuring the decomposability of a graph. This is accomplished by removing relationships until the graph is separated into subgraphs [152]. This is an extension to existing methods noted in their research.

Removing relationships until the graph is separated into subgraphs demonstrates the coupling found in the entity-relation subgraph, where a graph that is easily separable into distinct graphs is not highly connected. An algorithm manipulates the graph that is being examined in terms of connectivity. First, unary constraints are removed. These are relations that do not contribute directly to the connectivity complexity of the graph. The graph connectivity algorithm is then applied recursively against all resulting subgraphs generated during the process. A cumulative record, or score, is maintained to quantify the connectedness of the graph. A graph with low connectivity will have a low score. This procedure can be applied for each node, resulting in the appropriate values of ρ_i for use in Equation (9). Once the design variables and relationships are defined, this approach is readily automatable. Overall, the usefulness of this approach is readily evident, however, system architects engaged in capabilities-based design must ensure that the chosen design variables and relationships do not prematurely exclude portions of the design space that may later provide useful alternatives.

5.6.1.4 Historical Database Estimating

Both the CoBRA and COSYSMO cost estimating techniques previously discussed in Section 3.2 estimate the impact of complexity using historical databases of previous systems and projects. The complexity index developed in the CoBRA analogy based estimating technique and the COSYSMO CER developed using various size and cost drivers each provide a firm basis for developing ρ. Of course, developing ρ using methods such as these require a historical database that adequately reflects the problem at hand. For capabilities-based acquisition this may not always be the case, and system architects must take this into careful consideration.

5.6.1.5 Matrix-based Requirements Correlation Analysis

The final approach outlined in this section takes a slightly different approach to estimating ρ. This approach is primarily based on the limitations inherent in early phase, capabilities-based conceptual design. The principal assumption of this approach is that SoS-level performance requirements can be used in the calculation of FDC, rather than direct function-to-function comparisons, especially since the number of top-level SoS performance requirements are usually smaller and more manageable. This assumption is based on the following [65]:

- System functionality exists to support tasks that provide the desired SoS capability.
- SoS-level performance requirements specify the extent to which the SoS capability must be achieved.
- Each function should be traceable back to a SoS-level performance requirement or possibly multiple performance requirements.
- Relationships can exist between performance requirements.

Thus, there is a clearly defined link between system functionality and SoS performance requirements. Based on the strength of these relationships, as the number of SoS-level performance requirements increases and as the level of performance dictated by these requirements change, the nature of the interactions between functions will change as well. These relationships can be readily and compactly captured in a matrix format such as the one shown here:

$$R = \begin{bmatrix} 1 & r_{1,2} & \dots & r_{1,n} \\ r_{1,2} & 1 & \dots & \vdots \\ \vdots & & \ddots & \vdots \\ r_{1,n} & & \dots & 1 \end{bmatrix} \tag{15}$$

The matrix R is the Requirements Correlation Matrix. For an architecture with n number of performance requirements, R takes the form of a symmetric $n \times n$ matrix and uses either the estimated or measured correlation between requirements, $r_{i,j}$, to infer the amount of functional coupling that exists. Again, this is possible since requirements can be traced back to individual functions. Thus, $0 \leq r_{i,j} \leq 1$ except along the diagonals, where $r_{i,j} = 1$ for $i = j$. Thus, for independent, unrelated requirements $r_{i,j} = 0$ for all $i \neq j$. For highly correlated requirements, $r_{i,j} = 1$ for all i and j. In between are various loosely/tightly coupled matrices. Different options exist for determining the correlation between requirements. Subject matter expert opinion can be elicited, or the values can be obtained through (M&S). Since R is a symmetric square adjacency matrix with nonnegative entries, $\lambda^{(R)}$ is defined as the real, positive eigenvalue among the set of eigenvalues that is also the greatest in terms of absolute value [122]. Use of the matrix format results in the value of $\lambda^{(R)}$ ranging from a value of 1 for completely uncoupled requirements (irrespective of the number of requirements) to a value of n for completely coupled ones. Therefore, $\lambda^{(R)}$ can be directly substituted for ρ in the FDC formulation, if desired. Another advantage to this approach that makes it the preferred method for use in conceptual design is that the matrix format lends itself easily to automation. Finally, capturing the information in this format provides an easily traceable way for system architects to view the impact of changing requirements on architecture complexity.

5.6.2 Functional Process Complexity

When more than one system within the architecture is capable of carrying out a system level function, multiple paths are created through which the SoS can provide the intended capability. While this helps to reduce vulnerabilities and provides the system architecture with varying levels of flexibility, it also adds to architecture complexity. From an acquisition standpoint, more testing is required to determine and verify operation of the SoS and understand all the interactions that can occur. Potentially harmful interactions between systems must also be identified and ruled out, making system-to-system integration for highly complex architectures an important consideration during acquisition. Consequently, the coordination of these functions requires a separate accounting. This is captured by the FPC.

Conceptually, the arrangement of operational tasks and activities that must be executed in an ordered sequence to create a SoS capability is easily comparable in structure to a computer program. With this in mind, a program control graph similar to those used in graphically depicting the connections between blocks of code in a computer program can be adapted for use in describing military SoS architectures. An example program control graph (G) made up of n nodes (depicted as circles), e edges (depicted as arrows), and p connected components is shown in Figure 37. Each node in G is representative of a block of code in the program, "where the flow is sequential and the arcs correspond to branches taken in the program" [118]. G is defined in such a way that there are "unique entry and exit nodes, all nodes reachable from the entry, and the exit reachable from all nodes" [118]. Defining G in this way means that $p = 1$ unless G is composed of a hierarchical nest of subroutines.

The dizzying pace of advancements in the IT field over the years has forced software architects to quickly confront issues of growing design complexity. Hence, many techniques for quantifying and limiting software code complexity have been formulated. One of the most well-known measures is Cyclomatic Complexity, developed by

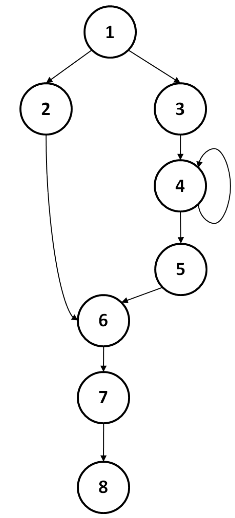


Figure 37: Example Program Control Graph.

Thomas McCabe in 1976. Cyclomatic Complexity (M) is a graph-theoretic complexity measure used to identify and control the number of paths through a program [118]. It is defined in terms of linearly independent basic paths. The combinations of these basic paths will generate every possible path through the program. M takes on values that are ≥ 1. $M = 1$ when there is only one linearly independent path through G. The equation for determining $M(G)$, the Cyclomatic Complexity of program control graph G, is then:

$$M(G) = e - n + 2p \tag{16}$$

When calculating M for a SoS architecture each node now represents a constituent system that is able to perform the necessary function. The program control graph subsequently becomes a depiction of the sequential flow of functionality through the systems that comprise the architecture. Care must be taken when calculating M, however, since a military SoS may have more than one entry and exit node, violating

McCabe's criterion. In this case, separate subgraphs are formed and must be considered individually. The M values from each subgraph can then be linearly combined to achieve the final M. Figure 38 provides example Cyclomatic Complexity calculations for alternative groupings of military systems performing a range of generic functions A-F. This example highlights how quickly the number of system-to-system interactions can increase due to slight changes in functional allocation and the number of systems included in the SoS architecture. In all, McCabe's Cyclomatic Complexity measure provides a single integer value that is easy to interpret, allows for direct comparison between various designs, and is easy to calculate in the conceptual design phase. Because it highlights the number of linearly independent system-to-system interactions, it also provides system architects with a good foundation to develop plans to conduct testing and integration to ensure SoS capability can be met satisfactorily.

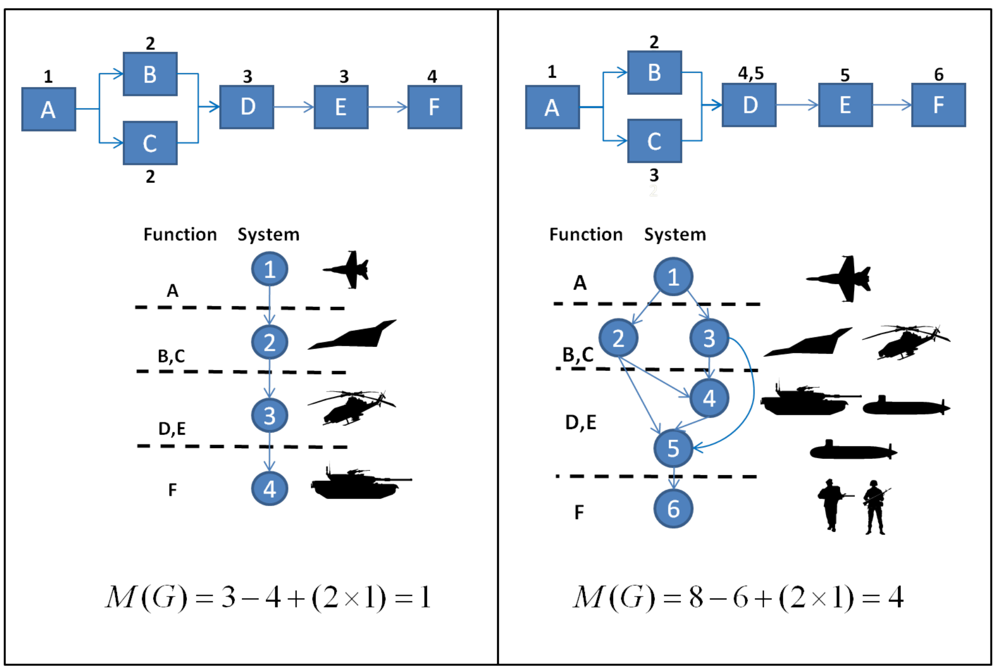


Figure 38: Example SoS Cyclomatic Complexity Alternatives [71].

An alternative to the path-based approach used to determine Cyclomatic Complexity is to take a modular approach when quantifying FPC. The modular approach also leverages the use of graphs to represent the logical flow of task/functional sequencing that defines a military SoS capability. Therefore, instead of counting the number of linear independent paths as a measure of the complexities that arise from potential variations in system-to-system interactions, an undirected graph can be partitioned using mathematical techniques. This will reveal different "clusters" of functionality, or modules, that represent a grouping of related functions that together perform a single logical task [65]. The undirected graph used in this analysis is based on the connectivity of different systems derived from the flow of functionality dictated by the program control graph.

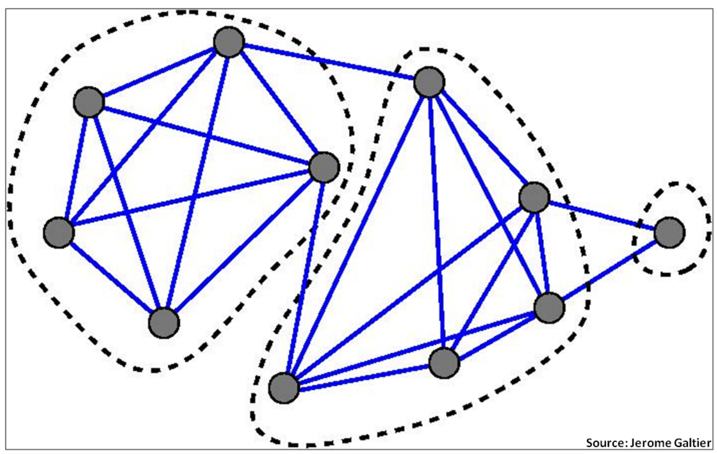


Figure 39: Example of a Partitioned Graph.

Different graph-theoretic options exist for partitioning graphs. For example, the Fiedler vector (FV) can be used. The FV is defined as the eigenvector associated with the second-smallest eigenvalue of the Laplacian matrix of G [21, 78]. Another useful option for partitioning an undirected graph is to determine the *strength* of an undirected graph. The graph strength, $\sigma(G)$, corresponds to the minimum ratio of the total strength of edges to the number of additional connected components created by deleting edges from the graph [89]. The strength of an edge is the nonnegative weight of the edge. For example, the graph shown in Figure 39, which is partitioned into 3 parts, has $\sigma(G) = 2$. These are just a sampling of the different options that graph theory provides. Ultimately, it is the choice of the system architect to determine the most suitable method to apply. Once the graph is partitioned into functional modules, McCabe describes an integration complexity measure based on the hierarchical relationships that exist among modules of a system [119]:

$$S_1 = S_o - n + 1 \tag{17}$$

First, the module design complexity, S_o, is calculated for each n number of functional modules by eliminating any complexity which does not influence the relationship between design modules. S_o for a given module depends on the design complexity of that module and any descendant modules that it must interact with. Then, using the top-level module, the integration complexity S_1 is measured. McCabe's integration complexity quantifies a basis set of integration tests, whose number is $S_1 \times n$, and S_1 is a function of the top module's design complexity and the number of modules.

5.6.3 Resource State Complexity

The focus of NCW is to foster the capability to "share, access, and protect information to a degree that it can establish and maintain an information advantage over an adversary" [11]. The nascent field of complexity science also recognizes that information processing/computation is an important aspect of complexity, especially when it involves the parallel or decentralized computation that can occur from multitudes of networked components [123, 163]. Within these networks, it is the aggregation of localized interactions that is the dominant, observable feature that gives rise to complex behavior. Because each system function within the SoS architecture can be associated with a set of localized information and service exchanges, it is logical to extend this concept to resource flows in general. Consequently, the interfaces that bind together a complex system are necessary to facilitate the flow of resources that provide unique, value-added SoS functions [68]. For purposes of this research, the word *interface* is used as a broad term that encompasses any interconnection that enables the flow of resources between systems. Thus, there are many types of interfaces. Perhaps the most familiar are those used in computing and communications-electronics systems. Interfaces can also encompass infrastructure, procedures, and people that allow resources to be exchanged. In a military SoS context, interfaces ensure that the needed capability can be achieved at the desired level of performance.

An interface possesses two aspects – both a needline and a technical implementation – that correspond to its logical and physical components. A needline documents the required or actual exchanges of resources between systems, and is a conduit for one or more resource exchanges – *i.e.*, it represents a logical bundle of resource flows. As previously shown in Figure 13, DoDAF depicts needlines in the OV-2 model. An additional OV-2 of a military Hunter-Killer UAV architecture is also presented in Figure 40.

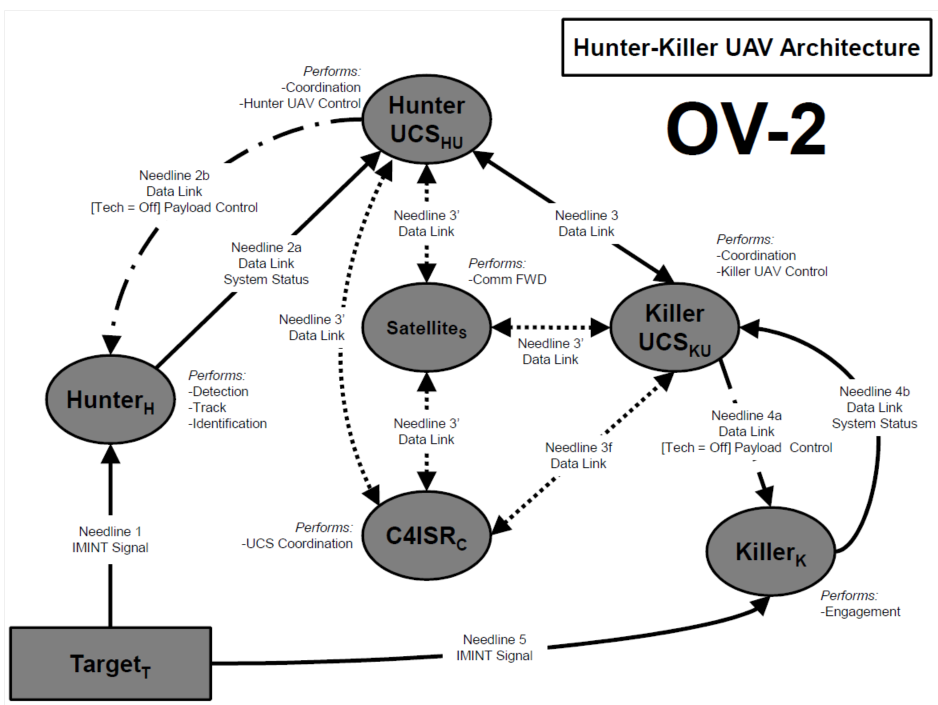


Figure 40: OV-2 Depicting Needlines for a Hunter-Killer UAV Architecture [21].

A needline can be either uni-directional or bi-directional. Needlines are useful in explaining which systems are involved in a resource exchange (*the who*) and the nature of the resources that need to be exchanged (*the what*). This is independent of the detailed implementation of the interface (*the how*), which is better described by technical considerations and design requirements [64]. Examples of these include those described in Ref. [100] such as quality, security, and reliability requirements.

Previous observations obtained from the study of Elementary CA revealed that classifying the complexity of system behavior means understanding the organization of the underlying patterns and sensitivities that govern these state transitions. Understanding the patterns can be accomplished at least in part by measuring the λ parameter and by describing the topology of the resource sharing network. Likewise, state transition sensitivity can be estimated by specifying μ. Extending these concepts to military SoS architectures means determining the key factors that affect how resource exchange occurs and its impact on the global state of the complex system. These factors will be primarily responsible for causing changes to the state and behavior of the SoS. In similar fashion to Elementary CA, complex systems such as a military SoS undergo state transitions. Each type of resource affects the SoS in different ways. This occurs as resources are exchanged via interfaces. The communication of information helps to change the state of the level of knowledge. For instance, a common operational picture may emerge. The transfer of energy & materials alters the physical parameters of systems, allowing functionality to be achieved. Examples include temperature, pressure, and fuel reserves. Lastly, the exchange of labor & expertise potentially influences both the level of knowledge as well as some physical parameters in the case of trained technicians. If each node within a complex system is considered analogous to a site in an Elementary CA, then estimating the sensitivity of the overall system to changes in resource processing at an individual site within a local neighborhood requires understanding the number of configuration changes that can occur at a particular site. As resources are being processed locally, a localized state transition at a site could propagate through to have global effects. Furthering this analogy to complex systems, a measure of the sensitivity of the overall system to these local perturbations requires characterizing the resource state space in terms of the number of significantly different system configurations it will allow.

The following example borrowed from Ref. [40] will help clarify. A computer network comprised of n number of nodes that is only capable of processing Top Secret classified information at all points in the network is characteristic of a homogeneous resource state space. While there are technical challenges to implementing the needed security protocols to handle classified material, there is little likelihood of interaction between classified material and unauthorized users within the system boundaries (providing that the integrity of the system boundary remains intact, of course). Thus, this system is only configured in one way and localized changes at each computer terminal will most likely have little effect on the overall state of system security. Likewise, a computer network only capable of processing Unclassified information can be characterized as a homogeneous resource state space as well. Figure 41 provides a visualization of each resource state space.

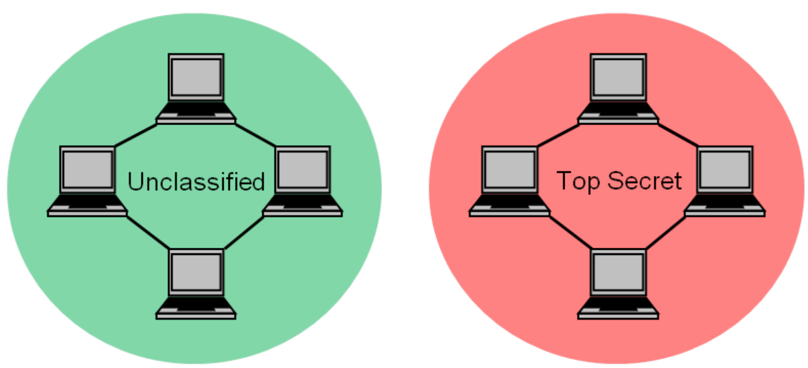


Figure 41: Example Homogeneous Resource State Spaces.

On the other hand, if the same system must allow access to personnel cleared for both classified and unclassified data, then the resource state space becomes more heterogeneous—there is an increased number of possible interactions at the local level between unauthorized users and classified data. This research assumes that this results in added complexity, since additional procedures, protocols, and resources must be put in place to describe and maintain system operations. Resource exchange is also affected as filters may be needed to prevent contamination of the unclassified portion of the network with classified data. Finally, since there is a greater number of significantly different configurations present, the system is also more sensitive, especially since changes in security policy at the local level could potentially have an effect on globally maintaining system security. Figure 42 provides an example of this.

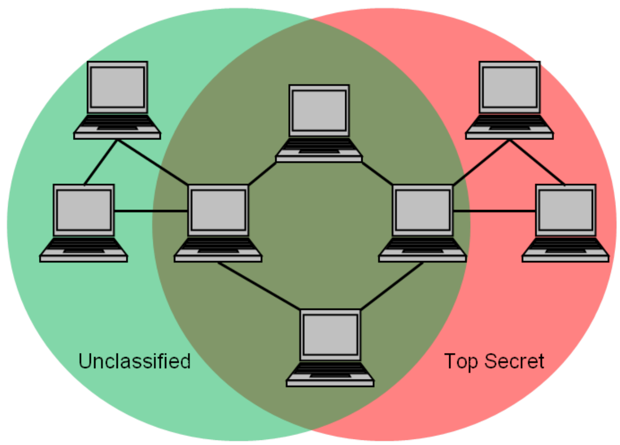


Figure 42: Example Heterogeneous Resource State Space.

Taking this into consideration, it is possible to define a set of Resource State Specifiers (RSS's) that characterize the nature of each needline within the resource state space. This is accomplished by taking a broad, somewhat abstract view of what different types of resources have in common in terms of affecting state transitions. Three RSS's in particular are defined for this research:

- Dimensionality
- Frequency
- Potential

Dimensionality is important because the more multidimensional the resource, the more information can be extracted from it. For example, many joint military operations require the use of Geospatial Intelligence, or GEOINT. The dimensionality of GEOINT products can range from basic aerial or satellite imagery to information products that create a common operational picture through the use of multiple and advanced sensors. This results in multiple types of data and information that can include operations, planning, and logistics that can be integrated into the GEOINT product [95]. The study of dynamical systems shows that as a system operates at multiple different time scales, perturbations and fluctuations may reach a point that affects returning to an equilibrium state, impacting system stability [155]. Finally, a resource's potential speaks to its influence, or probability to affect change. While this list may not be be exhaustive, the RSC measure that will be developed will be flexible in accommodating additional RSS's. Table 2 gives relevant examples of how to evaluate the different categories of RSS's for different types of resources.

Once the relevant RSS's are identified, the next step is to use them in developing a measure of RSC. Recall that it is the homogeneity vs. heterogeneity of the resource state that affects the sensitivity and thus complexity of the complex system. For a military SoS, this translates to identifying and classifying each needline according to the RSS's. To save time, redundant or identical needlines need not be included in the formulation, and the needlines may be grouped into categories deemed most appropriate by the analyst or system architect. Continuing with the computer network example, two types of needlines are identified and evaluated for the Top Secret homogeneous resource state space and the heterogeneous resource state space. The network provides users with both foreign country news & updates as well as domestic news & updates. Depending on the fidelity of the analysis, binary (0 or 1) values can be used to specify each RSS for a particular needline, or percentages can be used. In binary terms, a value of 0 means that the resource is always at or near the

Table 2: Resource State Specifier Examples.

RSS Description	Information Example	Energy/Materials Example	Skills/Labor Example
Dimensionality	Annotated imagery, maps with overlays, compiled report from multiple intelligence sources	Replenishment consisting of various fuel grades, resupply with multiple forms of ammunition	Service requires personnel from multiple divisions,organizations, multiple technical disciplines
Potential	Classification/Sensitivity such as Unclassified or Top Secret	Hazardous, volatile, energetic, or inert Importance to mission completion	Level of authority, training, or experience relevant to the task at hand
Frequency	UAV Payload control data continuously updated for real-time monitoring	Single fuel replenishment occurring in-theater during mission operations	On-site technicians required for full monitoring and maintenance

minimum/baseline level for that RSS, but not that it lacks any dimensionality, for instance. Conversely, a value of 1 means that the resource is always significantly more multidimensional than the baseline level. Percentages can be used to indicate the amount of variation if more fidelity is desired and if the information is available. Using frequency as an example, a value of 0.8 means that for a significant portion of time during operations, the frequency at which the resource is transmitted/received is at or near the highest rate. A value of 0.5 indicates that on average the frequency varies approximately equally in a range between the baseline and maximum values. More than one measure can be used for each RSS, as well. For example, both encryption and level of security classification could be used to measure differences in potential. Also, the types of systems that are involved may impact the resource state space and must be considered too. For example, an older computer on the network with limited processing may not be able to handle files encoded in certain multimedia formats. In terms of military SoS, an Intel Satellite may be able to deliver highly multidimensional GEOINT data to those who request it, but a mobile SOF team deployed in the field may be limited to downloading and viewing data in simpler formats. If this is the case, separate Resource State Characterization tables could be developed. In order to provide a simplified analysis, Tables 3 & 4 represent the network requirements set forth by system architects that each computer network alternative must adhere to.

Table 3: Top Secret Homogeneous Resource State Space Needlines.

Needline Description	Dimensionality	Frequency	Potential
Foreign Country News & Updates	1	1	1
Domestic News & Updates	1	1	1

Table 4: Homogeneous Resource State Space Needlines.

Needline Description	Dimensionality	Frequency	Potential
Foreign Country News & Updates	1	1	0.8
Domestic News & Updates	1	1	0.3

If the resource state space is comprised of more than one needline, the resource state space is automatically heterogeneous. If more than one needline exists, the following equation can be used to obtain the average of the values that comprise the resource state space, where l is the number of needlines and r represents the number of RSS categories.:

$$\gamma = \begin{cases} 0 & \text{if } l = 0, 1 \text{ (Num. of Needlines = 0 or 1)} \\ \frac{1}{(r \times l)} \sum_{i=1}^{l} \sum_{j=i}^{r} RSS_{ij} & \text{if } l \geq 2 \text{ (Num. of Needlines} \geq 1\text{)} \end{cases} \tag{18}$$

For the Top Secret homogeneous resource state space $\gamma = 1$. For the heterogeneous resource state space $\gamma = 0.85$ is obtained. Recall that a homogeneous resource state space occurs if all of the values in Table 12 are either 0 or 1, resulting in averages of 0 and 1, respectively. As the average move closer to 0.5, the character of the resource state space is less homogeneous and more heterogeneous, with an assumed increase in complexity. As a result, determining the level of homogeneity/heterogeneity of the resource state space can be accomplished in a straight–forward manner by applying Shannon's measure of information entropy in the case of two possibilities with probabilities p and $(1 - p$ [147]. The use of Shannon's entropy is found in many studies of complexity, including the study of Elementary CA's [82, 104, 107], and proves especially useful in this context. In a sense, what we are trying to measure is the entropy of the resource state space, or the global effects on the macrostate of the complex system due to the distribution of the microstates of individual nodes engaging in resource sharing and processing with their neighbors. The plot of Shannon's information entropy for this particular case is given in Figure 43. The plot is generated using the following equation with a logarithmic base of 2 used:

$$H(x) = -\sum_{i=1}^{n} p(x_i) log_b p(x_i); \{x_i : \text{i} = 1, \ldots, \text{n}\} \tag{19}$$

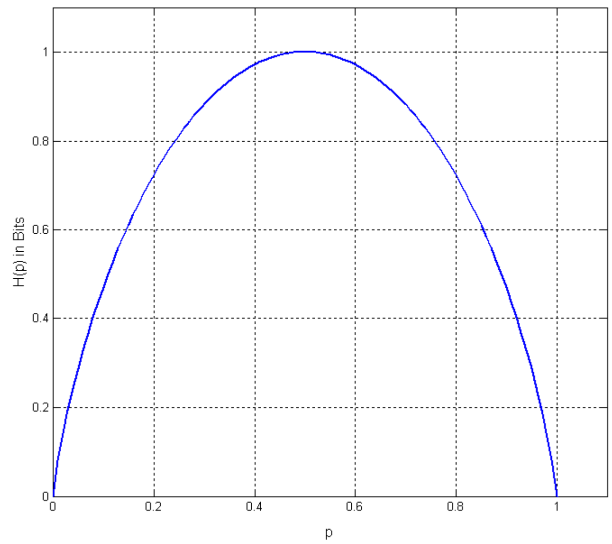


Figure 43: Entropy In the Case of Two Possibilities With Probabilities p and $(1-p)$ [147].

Since $b = 2$ is used as the logarithmic base to produce Figure 43, the resulting $H(p)$ is measured in bits. So if γ is substituted for p, $H(\gamma) = 0$ for the Top Secret homogeneous resource state space. In similar fashion, $H(\gamma) = 0.6$ for the heterogeneous resource state space. The remaining step is to combine the $H(\gamma)$ value obtained for each resource state space with the networked effects of uni-directional and bi-directional resource sharing. Networked effects are important because the network structure creates feed-forward and feedback linkages [42]. The same principle is used in the abstraction based complexity measure, but instead of measuring the number of cycles, graph energy is calculated instead.

As a reminder, the RSC sub-measure is meant to capture the sensitivity of the overall system to local perturbations as resources are being exchanged. Hence, the more homogeneous the resource state space and the less connected each node, the less likely local perturbations will develop and spread in a manner quickly enough to significantly disrupt system behavior. This should result in a low RSC score.

Extending this logic, it follows that a high RSC score will be indicative of a resource network with high connectivity and a heterogeneous resource state space. Graph theory provides a way to model the resource network in matrix form, allowing the use of spectral analysis to study the properties of the graph. There are numerous measures associated with spectral graph theory and many researchers have conducted extensive studies. Within the problem domain of military M&S, both Cares and Balestrini-Robinson provide excellent reviews of the usefulness of many of the existing measures in analyzing the fundamental properties of military networks [21, 42]. In general, the measures can be divided into those that help characterize vertex centrality and those that represent more global network properties. Centrality is the measure of the importance of a specific vertex (node) within a graph relative to other vertices. For the purposes of calculating RSC, more attention will be paid to measures that characterize the connectivity of the network on a more global scale, since we are interested more in how fast and far localized perturbations can spread to cause system-wide disruptions. There are numerous measures that exist, but this research will focus on a few primary ones. The following is an abbreviated list to highlight the options available to system architects [19, 21, 28, 42, 122]:

- *Link/Node Ratio*: Compares the link densities of different networks.
- *Perron-Frobenius Eigenvalue* (PFE): PFE is simply the largest, non-negative eigenvalue that exists for an adjacency matrix, where the adjacency matrix is assumed to be a sparse non-negative matrix.
- *Coefficient of Networked Effects* (CNE): CNE estimates the networked effects per node. CNE is the PFE normalized for network size by dividing the PFE by the number of nodes. CNE focuses on the creation of sub-networks, where a sub-network defines the patterns of localized interaction and collaboration that occur when the resource flow between nodes describe a path that revisits at least

one node once. If there are no cycles in a network, then no useful networked function is completed [42]. If the resource flow is incomplete and the required sub-networks do not exist, a zero value is obtained. As additional nodes are added that increase the number of sub-networks, the measured value increases as well.

- *Characteristic Path Length* (CPL): The median of the average distance from each node to every other node in a network. A short CPL means that resources can proliferate through the network without passing through a high number of nodes. For a network comprised of n vertices or nodes, CPL is calculated as $\log(n)$. Path horizon, which is a measure of how many nodes, on average, a node must interact with for self-synchronization to occur is listed as a separate measure but is also calculated as $\log(n)$.

- *Clustering*: A measure of the local cohesion in a network. The clustering coefficient is the ratio of the number of actual links between neighbors to the number of possible links between neighbors. Highly clustered networks tend to have pockets of connectivity, which can increase the connectivity and redundancy of the entire network.

- *Graph Energy*: Graph energy is defined to be the sum of the absolute values of the eigenvalues of the associated adjacency matrix. Graph energy represents the density of connections in the system with respect to the total number of possible interconnections. It is intended to be a measure of the dynamic instabilities that can increase as both systems and time scales become more interconnected.

- *Algebraic Connectivity*: The eigenvector associated with the first non-trivial eigenvalue, or the second-smallest eigenvalue, of the Laplacian matrix is commonly referred to as the Fiedler vector. The Laplacian matrix is constructed using the adjacency matrix and a matrix of vertex degrees. The degree of a vertex is a count of the number of edges incident to that vertex. The associated eigenvalue is commonly referred to as the algebraic connectivity. The algebraic connectivity is greater than zero if and only if the graph is connected. The algebraic connectivity increases as the graph grows in connectivity. Algebraic connectivity provides a measure of the synchronizability of the network and is closely related to the CPL.

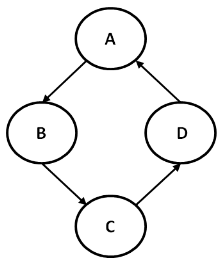
Use of these measures is analogous to determining the neighborhood size in the case of CA's, since the more cycles on average a node takes part in, the more likely that localized instabilities can propagate through the network. The use of eigenvalue-based measures such as the PFE, CNE, and Graph Energy requires defining an adjacency matrix for each alternative, taking into account the directionality of the needlines. This adjacency matrix is now defined as the RS matrix.

$$RS = \begin{bmatrix} 0 & rs_{1,2} & \dots & rs_{1,n} \\ rs_{1,2} & 0 & \dots & \vdots \\ \vdots & & \ddots & \vdots \\ rs_{1,n} & & \dots & 0 \end{bmatrix} \tag{20}$$

RS takes the form of a symmetric $n \times n$ matrix with main diagonal entries $=$ 0. To determine algebraic connectivity, the Laplacian matrix must be constructed instead. The normalized Laplacian matrix, $L = (l_{i,j})_{n\times n}$, for a directed graph with n vertices can be determined using:

$$l_{i,j} = \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{\sqrt{deg(v_i)deg(v_j)}} & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where either indegrees or outdegrees from each vertex can be used, depending on the application. Figure 44 shows the difference between the adjacency and Laplacian matrices for a simple network. Table 5 summarizes the results from analyzing the simple network in Figure 44 using the aforementioned network measures. Also, scores are calculated in the case where the edge/link between vertices B & C is removed, resulting in an acyclic network.



$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Adjacency Matrix

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 1 & -1 & 0 & -1 \\ -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \end{pmatrix} \end{array}$$

Normalized Laplacian Matrix (Indegree)

Figure 44: Example Adjacency and Laplacian Matrices For a Simple Network.

Table 5: Summary of Network Measures For a Simple Network.

Network Measure	Cyclic Score	Acyclic Score
Link/Node Ratio	1	0.75
PFE	1	0
CNE	0.25	0
CPL	0.60	0.60
Clustering	0.67	0.42
Graph Energy	2	0
Alg. Connectivity	1	0.38

Analysis of Table 5 shows that the network measures that are not eigenvalue-based will yield non-zero values for an acyclic network, complicating their inclusion into the RSC sub-measure. These measures include the Link/Node Ratio, CPL, and the Clustering Coefficient. Even though the Algebraic Connectivity is eigenvalue-based, vertex degree is also used in the construction of the Laplacian matrix. PFE, CNE, and Graph Energy prove to be useful measures, since they yield zero values for acyclic graphs and are able to capture the formation of sub-networks. As an example, Figure 45 shows 3 different variations of a combat cycle consisting of a sensor (S) that detects a target, a C2 node (C2) that makes a decision to engage the target, an engagement node (E) that engages the target, and the target node (T) [17]. For each combat cycle, the PFE of the corresponding adjacency matrix is calculated.

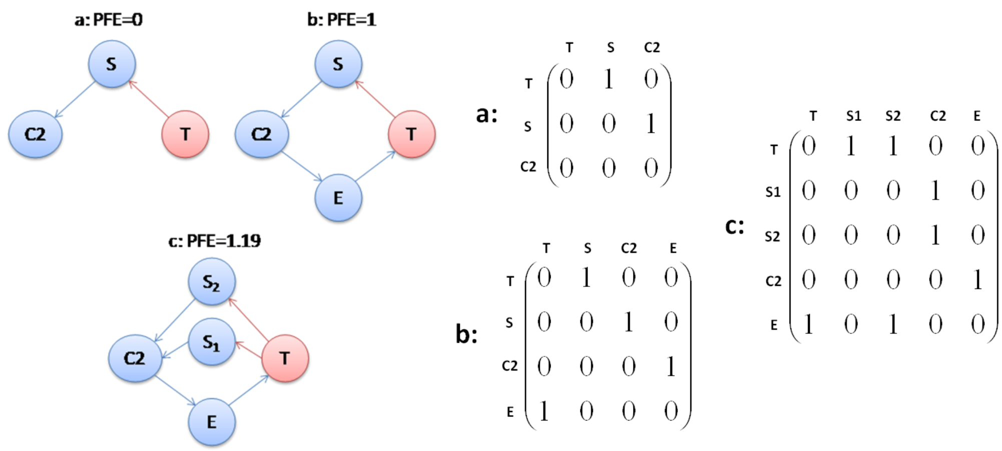


Figure 45: Measuring Networked Effects for Combat Networks.

The next step is to include the network connectivity score into the RSC sub-measure. Though Graph Energy can be used as well to yield similar results, CNE is the preferred method since it provides a direct accounting of the networked effects per node, and will be used in the following demonstration. Thus, the absolute maximum positive eigenvalue of RS, or $\lambda^{(RS)} = PFE$, can be divided by the total number of nodes ($N = n$) to obtain the CNE. Note that N depends on the force structure in place. Consequently, it is representative of the the total number of systems taking part in resource exchanges, since multiple instances of the same system type may be present. Since the diagonals of the RS matrix are defined as zero entries, a modified form of the CNE can also be used, if desired, where the PFE is divided by $N = n - 1$ nodes. This way, a value of 1 can be obtained for a fully connected, fully bi-directional resource network. Now, RSC can be calculated using the following:

$$RSC = [1 + H(\gamma)] \left(\frac{\lambda^{(RS)}}{N} \right) \tag{22}$$

where:

$$RSC = \begin{cases} 0 & \text{if } CNE = 0 \\ CNE & \text{if } H(\gamma) = 0 \text{ \& CNE} > 0 \end{cases} \tag{23}$$

This formulation yields $0 \geq RSC \leq 2$. Combining terms in Equation (22) yields an alternate form of the equation:

$$RSC = \mu\lambda^{(RS)} \tag{24}$$

In Equation (24), μ is defined as the Resource State Multiplier (RSM). For an acyclic resource network, RSC is zero since $\lambda^{(RS)} = 0$. This can be seen in Figure 46 for the computer network example problem. If the resource state space is homogeneous the contribution to architecture complexity due to network sensitivity is limited to the CNE, or connectivity per node. A cyclic resource network for the computer network example is shown in Figure 47. Table 6 shows the resulting RSC values for the different combinations of resource state spaces and resource networks for the computer network example. The fully connected computer network with the heterogeneous resource state space is evaluated as the most complex in terms of RSC, while the acyclic computer network is always zero complexity.

Table 6: Computer Network Example Resource State Complexities.

Heterogeneity: $H(\gamma)$	Connectivity: CNE	Complexity: RSC
0	0	0
0	0.4	0.4
0.6	0	0
0.6	0.4	0.6

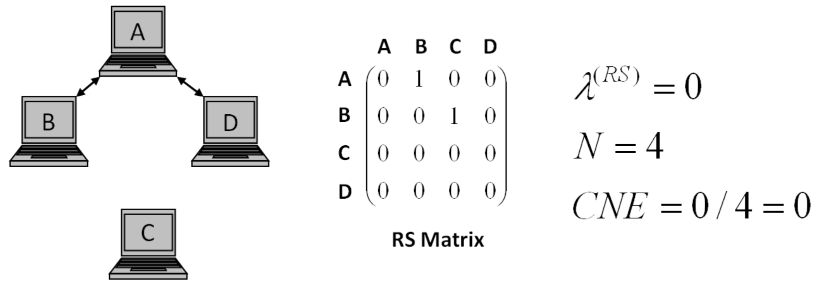


Figure 46: Acyclic Computer Network Example.

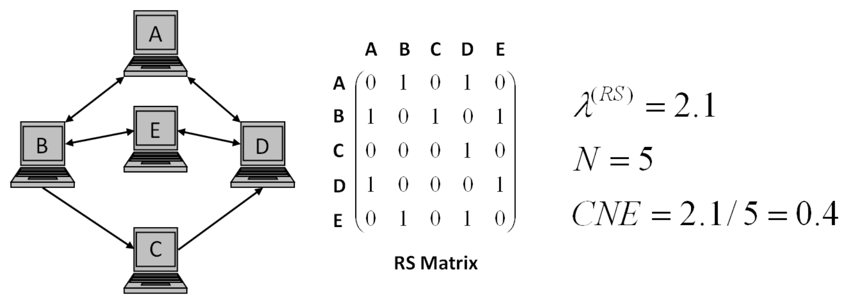


Figure 47: Cyclic Computer Network Example.

5.6.4 Resource Processing Complexity

While RSC captures important qualities of the needlines that make up the resource state space, it does not address the effects that the quantity or volume of resources plays in defining the complexity of the system. The study of elementary CA's shows that both the *amount* of information that is being exchanged among nodes within the system and the patterns of collaboration play a critical role in the dynamics of interconnected systems. Extending this logic to resource flows, RPC seeks to capture these aspects for military SoS. For example, the amount of resources exchanged between military SoS can vary based on such factors as the scope of the mission to be carried out, the level of performance required, or the available infrastructure. Different options exist for measuring resource quantity, depending upon the amount of information available to the system architect. Since there is usually limited knowledge during the conceptual design phase, for military SoS, a useful way of determining this is to specify the required level of collaboration between systems by stipulating interoperability. Interoperability is defined as [94]:

1) The ability to operate in synergy in the execution of assigned tasks.

2) The condition achieved among communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users. The degree of interoperability should be defined when referring to specific cases.

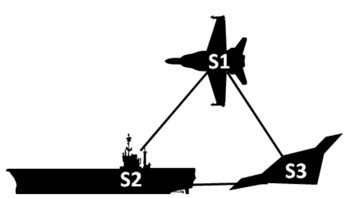
For military SoS, the North Atlantic Treaty Organization (NATO) specifies Standardization Agreements, or STANAGs, to help member countries define interoperability for a wide range of equipment and procedures. NATO STANAG 4586, for example, provides a five-level interoperability scale for the X-47B unmanned aerial vehicle (UAV) currently in development. This scale is provided in Table 7, which also shows how the concept can be generalized in order to make it more broadly applicable during the conceptual design phase. The interoperability levels (IOLs) put forth in Table 7 describe a hierarchy of interoperability. For instance, specifying an IOL of 3 between a pair of systems indicates that the resource exchanges might also include those that occur at IOL 1 & 2 as well. At this point, a subtle distinction should be made. Care should be taken to avoid confusion between performing an operational function from providing services in order to interoperate. For example, if System A is only tasked with deploying or retrieving another system with which it has no interfaces to enable coordination, control, or monitoring between the two, then System A is merely executing a function. System A is not operating in synergistic collaboration with that system.

Table 7 can be used to develop a weighted network graph to specify the interoperability relationships between systems. The associated adjacency matrix is the resource processing matrix, or RP. RP possesses similar attributes to the RS matrix, except that RP must be a symmetric matrix since interoperability defines a

Table 7: Military SoS Interoperability Hierarchy Levels.

Level	Generalized Description	X-47B (STANAG 4586) Example
0	Isolated or no exchange	
1	Indirect receipt/transmission of primary operational data, auxiliary ISR data	Indirect receipt/transmission of UAV-related payload data
2	Direct receipt/transmission of primary operational data, auxiliary ISR data	Direct receipt of ISR data where'direct' covers reception of UAV payload data by the unmanned control system when it has direct communication with the UAV
3	Direct receipt/transmission of control & monitoring data/services of asset weapons, munitions, sensors only	Control & monitoring of the UAV payload in addition to direct receipt of ISR and other data
4	Direct receipt/transmission of control & monitoring data/services; operational resupply services provided to asset	Control & monitoring of the UAV, less launch and recovery
5	Deployment, retrieval services provided to asset	Control & monitoring of the UAV, plus launch and recovery

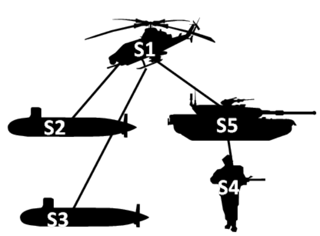
relationship that exists between a pair of systems. Similar to the RS matrix, the RP matrix can be evaluated using spectral graph measures such as eigenvalue analysis. Because RP is a symmetric matrix, the algebraic connectivity can be used as well. In general, the greater the number of systems that collaborate at high interoperability levels, the higher the RPC score should be. There are some architectural tradeoffs that exist between a small number of systems with high levels of interoperability vs. a large number of systems with low interoperability. The RPC sub-measure should reflect this. The IOLs for a group of systems can be used directly in creating the RP matrix, or normalized values can be used. Figure 48 provides an example RPC calculation for two different system groupings using eigenvalue analysis, where the $\lambda^{(RP)}$ value obtained from the PFE of the RP adjacency matrix is used directly for determining RPC. There is really no strong preference in choosing an eigenvalue-based analysis method in this case. The available methods that are presented are all simple to calculate and will yield similar results. The partial exception is algebraic connectivity, which requires slightly more computation to determine the Laplacian matrix. The PFE may be preferred by some simply because both CNE and Graph Energy require an additional calculation once the eigenvector is obtained.



$$\begin{array}{c c} & \begin{array}{ccc} \mathrm{S1} & \mathrm{S2} & \mathrm{S3} \end{array} \\ \begin{array}{c} \mathrm{S1} \\ \mathrm{S2} \\ \mathrm{S3} \end{array} & \begin{pmatrix} 0 & 5 & 3 \\ 5 & 0 & 5 \\ 3 & 5 & 0 \end{pmatrix} \end{array}$$

RP Matrix

$$RPC = \lambda^{(RP)} = 8.7$$



$$\begin{array}{c c} & \begin{array}{ccccc} \mathrm{S1} & \mathrm{S2} & \mathrm{S3} & \mathrm{S4} & \mathrm{S5} \end{array} \\ \begin{array}{c} \mathrm{S1} \\ \mathrm{S2} \\ \mathrm{S3} \\ \mathrm{S4} \\ \mathrm{S5} \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 2 & 0 \end{pmatrix} \end{array}$$

RP Matrix

$$RPC = \lambda^{(RP)} = 3.0$$

Figure 48: Example RPC Calculations.

IOLs between systems can change as mission requirements or operating environments change. This can force an increase in interoperability between systems that usually work together or emphasize the benefits of including optional exchanges. An optional exchange is defined as establishing a needline between systems that are not strictly required to interoperate from a functional standpoint, but their collaboration is expected to enhance SoS performance. Force structure represents the exact number of each system type (for example, the number of F/A-18's) that is available to participate in resource exchanges. In effect, for RPC numbers count. One item that is important to note is that the architect should only consider resource exchanges that take place within the operational timeline of providing the capability. Exchanges involving items that are logistic in nature typically operate on different time scales and should not be included in the interoperability assessment of the operational architecture. As a reminder, needlines do not represent the actual physical implementation of interfaces. This is especially important to remember when an IOL of 1 is specified, since in reality the resource exchange between the systems will take an indirect physical path. Determining the specific technical implementation of each interface is an exercise best left for a later, more detailed stage of design after the appropriate architecture has been selected. Figure 49 provides a visual example of how RPC translates to managing a variety of complex interfaces to facilitate resource exchanges for a particular architecture.

It is generally assumed that greater interoperability leads to increased collaboration, yielding positive effects such as improving the degree of shared awareness and force multiplication [12, 44, 68]. However, the opportunity to collaborate can also lead to negative effects as network connectivity increases and more resources are exchanged. Ref. [137] stresses that the most observable effect occurs as instances of information overload. This effect can be mitigated by enhancing network discipline, allowing enough processing time between different resource exchanges, or altogether

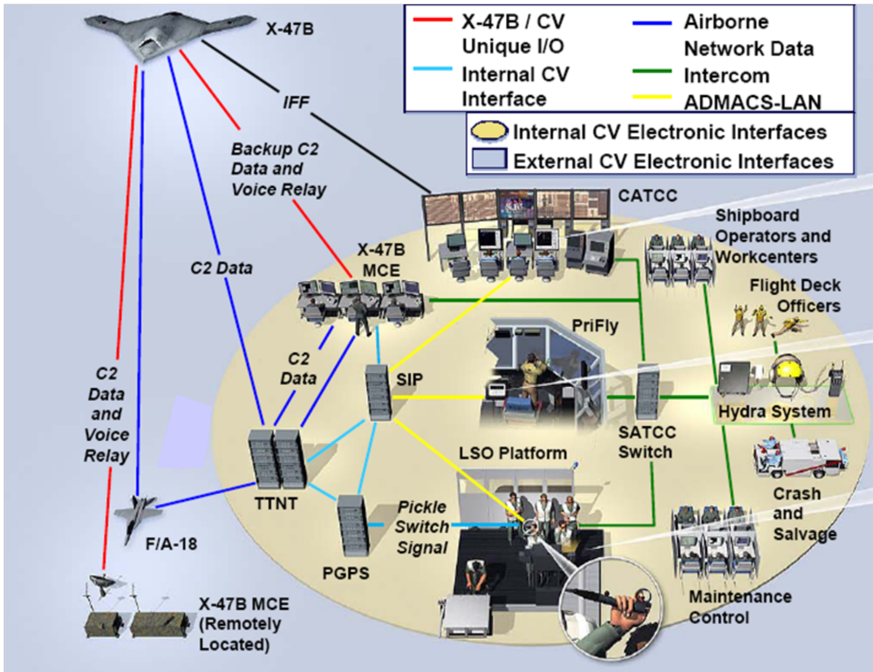


Figure 49: Example Network Interfaces For an Aircraft Carrier-Based UAV Architecture [49].

ignoring information or service demands that are deemed unimportant at the time. However, the fact that these measures must be put in place serves to emphasize the point that the patterns of interaction are influenced. Use of the RPC allows the system architect to evaluate the benefits derived from increased interoperability and force structure against the cost of complexity that will be incurred.

5.7 Defining the Measurement Framework

Figure 50 is a matrix of alternatives that summarizes the options available to system architects in developing the 4 relevant sub-measures to fully describe architecture complexity for a military SoS. The preferred methods for defining each sub-measure are highlighted in the figure. Now that methods exist for determining FDC, FPC, RSC, and RPC, the next step is to determine if the sub-measures can be combined into an overall measure of architecture complexity. The framework used to combine the sub-measures should not be chosen arbitrarily, especially since this affects the mathematical rigor and hence, the usefulness of the overall complexity measure. Two relevant theories, measurement theory and utility theory provide the necessary fundamentals to accomplish this.

5.7.1 Measurement Theory

The word *measure* has been used throughout this thesis. On the surface, the act of measuring appears to be a rather simple concept that is a fundamental aspect of physical existence. Yet the development of the theories and procedures surrounding measurement are rich with well-elaborated and differentiated methods and techniques that have taken centuries to acquire [30]. In formulating an architecture complexity measure, it will prove worthwhile to pause a minute and reflect on the insights, precise definitions, and concepts that comprise measurement theory. It is also necessary to reflect on exactly what desirable attributes the type of measure that is attempted in this research should possess in its application to architecture complexity.

The function of measurement, as described by Churchman & Ratoosh, is in "establishing metrical order among different manifestations of particular properties, and of making scientific events amenable to mathematical description" [48]. Measurement is often used to establish or clarify the relation between two different properties. Perhaps the most poignant description of the role measurement plays in science and engineering is to "connect two parts of theoretical knowledge, the mathematical and the conceptual, imparting relevance to the one and precision to the other" [48]. Therefore, the fundamental constituents of measurement as it applies to this research are the *object* on which a certain operation will be performed and the *observable properties* of the system whose "value" will be determined by this operation [30].

While there is scientific debate regarding the best classification scheme of different measurement scales, there are four widely recognized classifications of scale types. Table 8 provides a summary of these scale types [30, 48]. A fifth scale type, the

Architecture Feature	Measurable Trait	Sub-Measure
System Physical & Functional Boundaries	Organization & distribution of functionality among systems	Functional Distribution Complexity (**FDC**)
Functional Process/Task Sequence	Patterns of system-to-system interactions	Functional Processing Complexity (**FPC**)
Resource State Characteristics	Properties of & effects from the resources being exchanged	Resource State Complexity (**RSC**)
System-to-System Interfaces	Patterns of collaboration and resource sharing	Resource Processing Complexity (**RPC**)

Sub-measure	Alternative Analysis Methods				
FDC	**Node-level Functional Requirements Based**			**Top-level Performance Requirements Based**	
	1. Direct Function-to-Function Comparisons	2. Uncertainty-based Analysis	3. Graph-based Variable Analysis	4. Historical Database Estimating	5. Matrix-based Requirements Correlation Analysis
FPC	**Path-based Complexity**	**Module-based Integration Complexity**			
	1. Cyclomatic Complexity	2. Fiedler Vector Partition	3. Graph Strength Partition		
RSC	**Eigenvalue-based Adjacency Matrix Analysis**				
	1. Perron-Frobenious Eigenvalue	2. Coefficient of Networked Effects	3. Graph Energy		
RPC	**Eigenvalue-based Analysis** **Laplacian Matrix**		**Adjacency Matrix**		
	1. Algebraic Connectivity	2. Perron-Frobenius Eigenvalue	3. Coefficient of Networked Effects	4. Graph Energy	

Figure 50: Alternative Methods of Developing the Architecture Complexity Sub-Measures.

logarithmic interval scale, is widely recognized as well, but less popular than the linear interval scale. The logarithmic interval scale belongs to the exponential or power group. The mathematical description for this scale is $x' = kx^n$ where the constants k and n are positive. In general, the unique properties of each scale make them suitable for certain operations and not for others. For example, taking ratios between numbers on the interval scale are not meaningful. However, if one is left with equal, but unknown ratios the logarithmic interval scale can be used [48]. Also, some statistical operations are allowed with some scale types but not with others. The only permissible measure of location for a nominal scale is the mode, whereas nearly all statistics are applicable to measurements made on ratio scales, including the mean, median, mode, standard deviation, correlation, regression, and analysis of variance. This makes the ratio scale the most informative of the four. This is the fundamental reason why most measurement in the physical sciences and engineering utilize ratio scales [30, 48].

Since the aim of this research is to develop a complexity measure that *quantifies* how much more or less complex an alternative is relative to another one, this makes both the nominal and ordinal scales poor candidates. Intuitively, negative complexity scores are not desired either, further eliminating the interval scale. Therefore, the framework that will be developed will make use of the ratio scale. This requires specifying a non-arbitrary origin or zero point that has some meaning and significance in the context of military SoS architecture complexity.

5.7.2 Multiattribute Utility Functions

Utility is the assigning of a real number to a set of options such that a comparison of the real numbers will reveal the decision maker's set of preferences [92]. As seen with measurement scales, the principles of ordinality and cardinality apply to utilities as well. The use of utility curves/functions is a common methodology used in the

Table 8: Classification of Scales of Measurement.

Scale	Basic Empirical Operations	Examples
Nominal: Permutation Group $x' = f(x)$ 1-to-1 subst.	Determination of equality	Assignment of labels: *e.g.*, "Numbering" of football players, Categories of rocks (igneous, sedimentary, and metamorphic)
Ordinal: Isotonic Group $x' = f(x)$ $\uparrow$ monotonic function	Determination of greater or less	Rank Order: 1st, 2nd, 3rd, etc., "Bad", "medium", "good"
Interval: Linear Group: $x' = \alpha x + b$ $\alpha > 0$	Determination of the equality of intervals or of differences	Arbitrary zero point, negative values allowed: Temperature (Fahrenheit or Celsius), Position, Time (calendar), Energy (potential)
Ratio: Similarity Group: $x' = cx$ $c > 0$	Determination of the equality of ratios	Possesses a natural origin/ zero point: Length, density, work, Temperature (Rankine or Kelvin)

DoD and industry to perform tradeoff analysis. In the DoD it is widely used for cost-effectiveness analysis and proposal evaluation [65]. Utility curves can capture different relationships. For example, the relative utility of an attribute or decision factor can be represented as a constant value relationship (straight line), increasing value (concave curve), decreasing value (convex curve), or a stepped value. In addition, thresholds can also be applied. Figure 51 provides an example and highlights the difference between a step function and a continuous relationship.

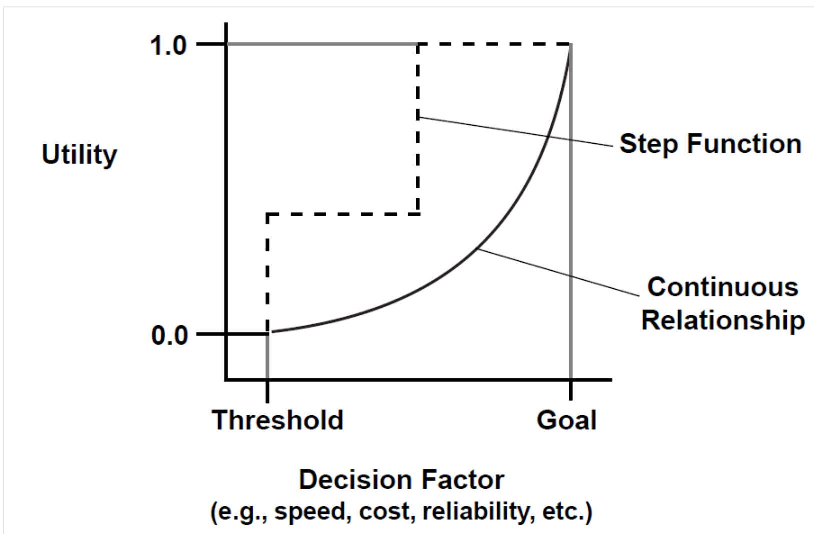


Figure 51: Continuous & Step Function Utility Curve Examples [65].

The use of utility theory can yield many benefits if used properly, and Hazelrigg is a proponent of using utility theory to develop properly formulated objective functions in the engineering design process [92]. The insights provided by utility theory can be leveraged to combine the individual sub-measures properly into an overall measure of architecture complexity. In particular, there are three relevant transformations that may be applied. They are the linear additive utility, multiplicative utility, and log-linear utility. The linearly additive utility function is a weighted sum approach. It is a conceptually simple and easy to use approach that is often employed. While this approach may be the most commonly used, according to Hazelrigg, it may also be the most misused. The use of the linear additive utility form would yield the following form for measured architecture complexity:

$$C_\alpha = w_1 FDC + w_2 FPC + w_3 RSC + w_4 RPC \tag{25}$$

where w_1–w_4 are weighting factors. For this specific application, the weights are necessary in order to counteract the effect of the magnitude of a single measure(s) overshadowing others and driving the combined score, which is often a concern when using linear combinations. Moreover, there is another equally important drawback that arises in using this formulation. It is readily observable that non-zero architecture complexity scores could be obtained even when the criteria established by the definition of a complex system are not met. For example, a system with nodes performing functions independently while there is no resource sharing or processing between them would have RSC and RPC scores of zero, but the FDC and FPC scores would not be affected. Thus, C_α would be equal to the weighted sums of FDC and FPC. This does not satisfy our preferences, especially since we wish to specify an origin according to the ratio measurement scale. If multiplication of the sub-measures is carried out instead, the architecture complexity score takes the following form:

$$C_\alpha = c_p \times (FDC \times FPC \times RSC \times RPC) \tag{26}$$

where c_p is an arbitrary positive constant. As Hazelrigg notes, setting $c_p = 1$ is perfectly acceptable. There are many benefits to using this form. First, the weighting problems inherent in linear combinations can be avoided. Since the distributive property holds for multiplication, individual sub-measures can be scaled using multiplier constants if desired without affecting the rank order, as long as consistency of the scaling factors is maintained when comparing different alternatives. Also, when making comparisons, a baseline score can be selected and all scores can be referenced by division to the baseline. Furthermore, $C_\alpha = 0$ for a single system in isolation and when an acyclic resource sharing network exists, or when there is zero interoperability between all systems *i.e.*, both RSC & RPC are zero. Based on the established definition for a complex system, this more closely matches expectations for complex systems/SoS at the origin, helping meet the criteria for a ratio measurement scale. Since Hazelrigg proves the equivalence of the log-linear, multiplicative, and linear additive forms given the proper choice of utility function, multiplicative utility proves a satisfactory form on which to base the final formulation of the architecture complexity framework [92]. If later desired, a proper utility function can be chosen such that the log-linear form can be used as well.

Hazelrigg states that a common mistake made by engineers is that the attributes themselves are typically used in the utility formulations, rather than converting the attribute to a utility. Failure to do this might result in the utility function being valid only in a small region, making them invalid when there are large changes in the attributes [92]. This prompts the re-evaluation of each of the sub-measures to ensure they adequately reflect intended preferences, especially since each sub-measure will be multiplied by each other. The FDC, RSC, and RPC, for instance, all seem to satisfactorily transform the system attributes they are intended to measure. The FPC however, requires additional consideration, especially where Cyclomatic Complexity (M) is concerned. First, practically, a SoS architecture with $M = 2$, or two linearly independent functional paths, is not necessarily twice as complex than a SoS architecture with $M = 1$. It just means there is one extra set of system-to-system interactions to evaluate. Secondly, the level of functional redundancy can be quite large, especially for a SoS comprised of tens of constituent systems. This could lead to very large Cyclomatic Complexity scores that is not necessarily representative of the true level of complexity. So in this instance, a transformation should be applied so that increasing Cyclomatic Complexity more closely reflects desired preferences and intuitive notions. Since the Cyclomatic Complexity scores can encompass a wide range of values starting at 1, a logarithmic transform makes the most sense. Log-linear scaling can be used to perform the scaling, so that $FPC = 1 + \log M$. For architectures with $M = 1$, the FPC yields a value of 1.0. For architectures that have a Cyclomatic Complexity of 2 and 4, FPC is 1.3 and 1.6, respectively. It is not until $M = 10$ that the FPC will score will reach a value of 2 and cause a doubling of C_α, with all else being equal. The following equation provides an example application of the multiattribute utility function, resulting in the particular formulation of C_α:

$$C_\alpha = \frac{\sum_{i=1}^{T} F_i^{\lambda^{(R)}}}{T} \times (1 + \log M) \times \mu\lambda^{(RS)} \times \lambda^{(RP)} \tag{27}$$

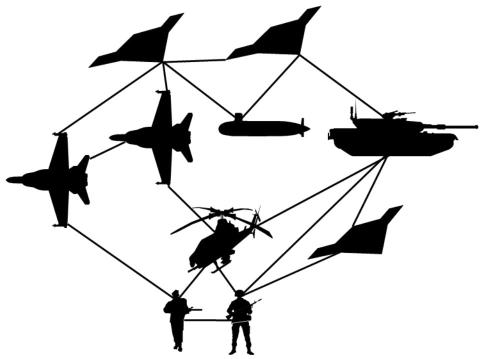
Figure 52 is an example of the lowest scoring feasible architecture that results from the use of Equation (27). Figure 53 shows the possible difference in scoring that can be obtained when a highly networked architecture with high functionality and a heterogeneous resource state space is evaluated.

$$FDC = \frac{1^2 + 1^2}{2} = 1$$
$$FPC = 1$$
$$RSC = [1 + 0] \times \left(\frac{1}{2}\right) = 0.5$$
$$RPC = 1$$
$$C_\alpha = 0.5$$

Figure 52: Example Architecture Complexity Score Obtained for the Lowest Scoring Feasible Architecture.

5.7.3 Sub-measure Independence

To make proper use of the multiattribute function it is important to prove the independence of the submeasures [92]. There is a natural division between the four measures, since two of them focus on measuring attributes of the system related to the organization of functionality within the architecture while the other two focus



$$FDC = 27$$
$$FPC = 1.7$$
$$RSC = [1+0] \times \left(\frac{1}{2}\right) = 1.8$$
$$RPC = 12.5$$
$$C_{\alpha} = 1{,}033$$

Figure 53: Example Architecture Complexity Score Obtained for a Highly Networked/Heterogeneous Resource State Space Architecture.

on patterns and characteristics of resource sharing that arise from needlines. It can be shown that FDC and FPC are independent of each other, for instance, using the following simple examples shown in Figures 54 & 55 for generic functions A-F. Figure 54 shows how FPC can remain the same even if the system's physical/functional boundaries shift. In this example, FDC will change as System 1's boundaries shift to include function D, resulting in functional overlap. If Cyclomatic Complexity is used to measure FPC, then the number of linear independent paths based on function sequencing is one in both cases. A modular approach to measuring FPC should yield the same results if the proper partitioning is conducted. Figure 55 gives an example of FDC of FDC remaining constant while FPC changes. Changes to FPC can occur in two cases. The first case is when bypassing of functions becomes allowable. This is the middle case in Figure 55 where the allowable bypassing of functions B-E means that another linearly independent path opens up between Systems 1 & 3. The second case involves trading functionality between systems, where in the bottom diagram Systems 1 & 3 trade performing functions B and F. If the difficulty associated with integrating these new sets of functions remains the same, then FDC remains unchanged as well.

The independence between RSC and RPC is based on the previously presented study of Elementary CA, where an independent phase parameter was developed to

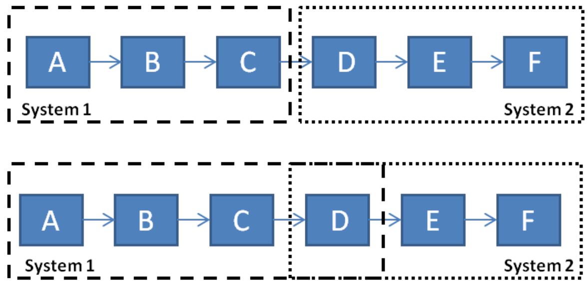


Figure 54: Example Showing Functional Processing Complexity Remaining Unchanged With a Change in Functional Distribution Complexity.

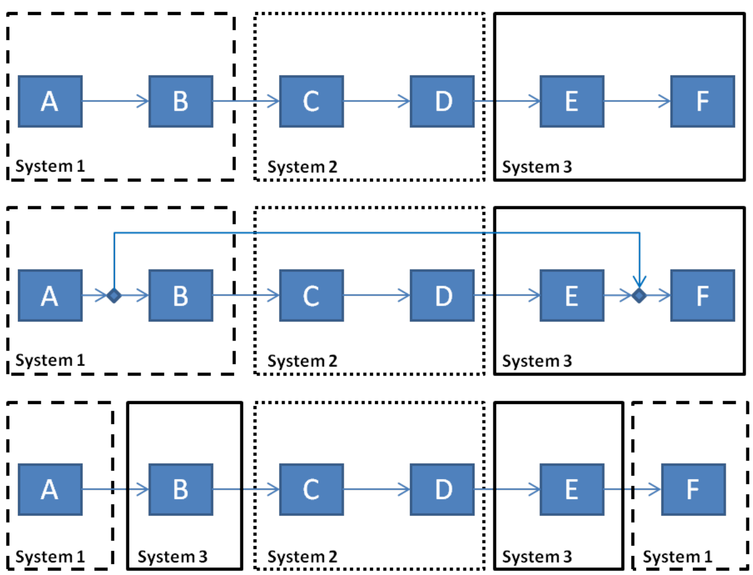


Figure 55: Example Showing Functional Distribution Complexity Remaining Unchanged With a Change in Functional Processing Complexity.

aid in the classification of different rules states. Thus, RPC focuses on measuring the level of interoperability between systems included within the architecture as a way of capturing the amount of resources required that must be processed by the architecture. RSC then serves to mimic the role of the independent phase parameter found in Elementary CA analysis, seeking to capture the sensitivity of the overall SoS architecture to small, localized perturbations. In this way, architectures with the same interoperability can be differentiated from each other based on the added complexity that can result as nonlinearities are introduced. These nonlinearities are due primarily to the characteristics of the resources that each alternative architecture is designed to process and the level of connectivity exhibited by the resource sharing network.

5.7.4 Architecture Complexity Measurement Framework

Now that the relevant complex system attributes, proper measurement scale, and most appropriate multiattribute function have been chosen, the measurement framework can be fully described. Equation (28) formally presents the resulting framework:

$$C_\alpha = [FDC \times FPC]_{\mathcal{F}} \times [RSC \times RPC]_{\mathcal{R}} \tag{28}$$

The framework presented in Equation (28) combines the fundamental aspects of complexity in such a way that should prove useful to system architects in identifying and communicating the relevant sources of complexity. C_α can be separated into two principal domains. The first domain, denoted by $\mathcal{F}$, represents the functional domain. $\mathcal{F}$ is primarily focused on the identification, organization, and allocation of the necessary tasks and activities that must be accomplished as well as the requirements that detail the extent to which functions must be executed. The second domain, labeled $\mathcal{R}$, represents the resource domain. Within $\mathcal{R}$ the primary concern is with the movement, sharing, and decentralized processing of resources. Resource exchanges are comprised of the transfer of useful information, energy & materials, and even skills & labor [11, 64]. The two domains are linked in that each system function typically requires some input of resources. Once a function is executed, any resulting resource that is output may be transferred to other systems in carrying out their functions.

In general, any means of quantifying complexity should meet the measurement criteria as put forth by McCabe & Butler [119]. It can be shown that the developed framework meets the McCabe & Butler criteria, and thus will prove useful to decision makers for comparing the architecture complexity of various alternatives. Based on the results presented in Figures 52 and 53, the framework can be used to develop a measure of architecture complexity that intuitively correlates with the difficulty of comprehending a design. Seemingly large, complicated design yield high C_{alpha} scores while simple designs are low scoring. By focusing on defining and using observable properties of complex systems, objectivity and mathematical rigor have been achieved. The third criterion is related to the effort to integrate the design. Inclusion of McCabe's Cyclomatic and design/integration complexity measures help capture the system-to-system interactions that provide end-to-end functionality. Also, the resource domain measures are useful in giving an indication of the complexities inherent in resource sharing and collaboration. Next, by taking an organizational approach to measuring complexity, the measurement framework lends itself readily for use in the conceptual design phase where there is often limited information. The ability to compute architecture complexity scores early in the acquisition cycle means that the measurement framework can be used to help generate an integration test plan early in the life cycle. Lastly, use of graph theory and associated analyses where appropriate serves to make the calculation of C_α easily automatable. This is especially important as the size of the design space grows to include large numbers of alternatives.

CHAPTER VI

DETERMINING ACQUISITION VALUE

Acquiring weapons systems typically requires the allocation of large amounts of resources to projects that develop a warfighting capability over a period of many years. For example, the GAO's ninth annual assessment of DoD weapons systems acquisitions showed that the fiscal year 2010 portfolio of major defense acquisition programs consisted of 98 programs totaling $1.68 trillion (FY2011) [87]. This makes strategic planning in order to determine the best allocation of resources an important factor [139]. A key element of a strategic plan is that it focuses resources on critical elements needed to meet the strategic objectives of an organization [148]. Also, a fluid, dynamic acquisition environment demands the strategic plan be flexible and adaptive to changing conditions.

The development of an architecture complexity measure suitable for military SoS means that now, architecture complexity becomes the primary currency of acquiring weapons systems, rather than relying on monetary cost estimates that have historically been proven inaccurate during the initial stages of design. Due to the time value of money, monetary costs usually must be discounted when making economic comparisons. Likewise, complexity costs must discounted as well. This discounting should be based on the procedures and processes in place to manage complexity. For instance, the most experienced personnel may be assigned to a highly complex project, schedules may be padded with additional slack time to cover unforeseen contingencies, or additional oversight may be instituted to maintain tight control over the planned interfaces between systems. A useful valuation framework must be able to take this into account while including uncertainty in the analysis.

6.1 Overview of Financial Valuation Methods

The fields of economics and finance have developed many methods for comparing the economic effectiveness of investment alternatives and opportunities. In the world of business, especially in a capitalistic market structure, studying the economic worth and desirability of an individual project or a portfolio of investments is central to developing a resource allocation strategy. Thus, various standard measures of economic effectiveness have been developed. They are Present Worth, Annual Worth, Future Worth, and Rate of Return (ROR). A brief summary of each is provided below [41, 106]:

- Present Worth: Involves the conversion of each individual cash flow to its present worth equivalent and the summation of the individual present worths to obtain net present worth.
- Annual Worth: Determined by converting all cash flows to an equivalent uniform annual series of cash flows.
- Future Worth: Obtained by converting each individual cash flow to its future worth equivalent and determining the net future worth for the project.
- Rate of Return: Among the many definitions of rate of return, the most popular definition is the interest rate that yields a net present worth of zero; such an ROR is referred to as the *Internal Rate of Return* (IRR).

The discounting of cash flows to account for the time value of money is a fundamental aspect of financial decision making. The mechanism used to express the time value of money is the interest rate, also called the discount rate and occasionally the opportunity cost rate [41]. Use of the interest rate can be seen in Equation 29. This type of analysis is commonly known as Net Present Value/Discounted Cash Flow (NPV/DCF) analysis.

$$NPV = \sum_{t=1}^{N} \frac{F_t}{(1+i)^t} \tag{29}$$

Where:

- i = effective interest rate per period
- N = number of compounding periods
- F_t = future sum of money at time t

Equation (29) relates present and future sums of a time-series cash flow, where cash is either received or expended in some repeating time interval (usually annually). When determining IRR, the most common method of calculation of the IRR for a single project involves finding the interest rate at which the present worth of the cash inflow equals the present worth of the cash outflow. Also, the IRR method involves comparison against a minimum attractive rate of return [41]. In terms of NPV, a project or investment is deemed worthwhile if a NPV of zero or greater is achieved. NPV/DCF assumes separate cash flows that are defined in advance [112, 58, 134]. Unfortunately, the main drawback of NPV is that it does not naturally account for the variance on the cash flow returns. Capturing this type of uncertainty is essential to decision making and must be explicitly included in the proposed valuation framework to obtain a proper determination of value. As this type of analysis relates to engineering design, in particular, Peoples & Wilcox make note that, "Value metrics, such as net present value (NPV), internal rate of return (IRR), and return on investment (ROI), have also been considered as design objectives. The problem has also been approached as a multiobjective optimization balancing cost and performance" [135]. They go on to note the shortcomings of these types of analysis when they state:

> Value metrics such as NPV are based on static valuations of the design. These metrics do not attempt to capture explicitly technical or financial

> uncertainties that may arise and, as such, do not properly account for the associated business risk of the program. Further, the related issue of flexibility—that is, the ability of the manufacturer to make decisions in response to unexpected or changing conditions—is not considered. In the field of finance, considerable research has been performed to develop more sophisticated valuation techniques to address the shortcomings of traditional valuation techniques. Substantial literature exists describing real options theory, which provides a way to quantify the value of a product or strategy in the presence of uncertainty.

This leads us to assess the viability of adopting Real Options for use as a decision-making framework for acquiring military SoS acquisitions. The principle difference that must be reconciled for this approach to prove useful will be the ability to monetize complexity. Providing an analysis that is independent from traditional financial metrics will help decision makers avoid focusing overly on reducing costs in the short term, to the detriment of the long-term evolution of the SoS architecture [135].

6.2 Real Options for Strategic Decision Making

When determining executive strategy, Luehrman also provides a poignant summary of the widely held shortcomings of NPV/DCF valuation [112]:

> When executives create strategy, they project themselves and their organizations into the future, creating a path from where they are now to where they want to be some years down the road. In competitive markets, though, no one expects to formulate a detailed long-term plan and follow it mindlessly. As soon as we start down the path, we begin learning—about business conditions, competitors' actions, the quality of preparations, and so forth—and we need to respond flexibly to what we learn. Unfortunately, the financial tool most widely relied on to estimate the value of

> strategy—discounted cash flow (DCF) valuation—assumes that we will follow a predetermined plan, regardless of how events unfold.

Thinking in terms of defense acquisition, where evolutionary acquisition strategies and rapidly shifting threat environments are encountered, there are obvious parallels that can be drawn. Since it is oftentimes difficult to get precise forecasts of the future business environment or projected cash flows, Luehrman goes on to suggest a better approach:

> A better approach would incorporate both the uncertainty inherent in business and the active decision making for a strategy to succeed. It would help executives think strategically on their feet by capturing the value of doing just that—of managing actively rather than passively. Options can deliver that extra insight. Advances in both computing power and our understanding of option pricing over the last 20 years make it feasible now to begin analyzing business strategies as chains of real options.

Therefore, incorporating elements of real options analysis into acquisition decision making would provide obvious benefits. In finance, an option is a contract giving its owner the right, but not the obligation, to buy or sell a fixed number of shares of a specified common stock or commodity at a fixed price at any time on or before a given contract date [51]. There are two types of options—call options and put options. A call option grants the right to buy the underlying security at the fixed price within a specified time frame while a put option grants the right to sell the underlying security at the fixed price within the given time frame. Therefore, while the price of the underlying security fluctuates, the option contract creates an agreed upon buy/sell price upon exercise. Thus, option contracts have a critical time component and their value is closely tied to an underlying financial security. Because of these features, financial options have many different uses by investors, for example risk management by using hedging techniques. Different methods of exercising options include American style, where exercise can be done at any time before expiration of the option, or European style where exercise is only available on the expiration date of the option. The method of valuation used by Luehrman in the development of the option space is the simpler European style option. American style options require more complex modeling methods to determine their price, such as either the Black-Scholes option pricing formula or the binomial option pricing formula.

The term 'Real Options' indicates that the options being considered now pertain to physical, tangible assets, such as equipment, rather than financial instruments [58]. The application of Real Options to engineering systems is a relatively new and growing endeavor, and their application in both producing and valuing flexible, adaptable engineering designs is a rapidly growing area of research [115, 58, 131, 134]. Real Options are also being used in projects in a multitude of ways. They can be used to determine the value of actions that either seek to grow and expand a project or to wait and defer action. Real Options also provide a means to value whether the best strategy going forward is to change direction mid-course or even abandon the project altogether. Advances in both computing power and our understanding of option pricing over the last 20 years make it feasible now to begin analyzing business strategies as chains of real options [112].

6.3 The Tomato Garden: Luehrman's Real Option Space

Figure 56 presents the option space as envisioned by Luehrman for comparing alternatives.

In his example, Luehrman uses the analogy that managing a portfolio of strategic options is like growing a garden of tomatoes planted in an unpredictable climate. To make clear the investment decisions for different alternatives, the options space is then subdivided into six different regions. As the gardener periodically checks

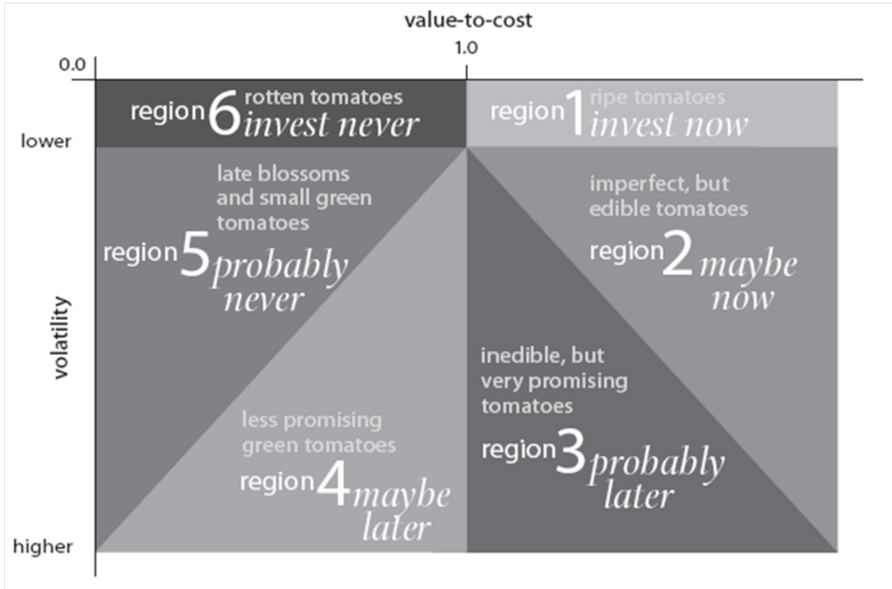


Figure 56: The Tomato Garden: Luehrman's Real Option Space [112].

on the status of his garden prior to harvesting, they will eventually find that some tomatoes are ripe and perfect and ready for immediate picking. This corresponds to the first region in the upper right-hand corner. At the other extreme would be tomatoes that have already gone rotten. These alternatives are grouped together in the upper left-hand region. Both of these regions have in common low volatility, i.e. their outcomes are relatively known and stable, but differ in their value-to-cost ratio. Region 1 tomatoes have value-to-cost greater than one while region 6 tomatoes are less than one, signifying their poor investment value. As Luehrman points out, these cases at the extremes-invest now and invest never-are easy decisions for the gardener to make. The regions in between are where tomatoes of varying prospect are, representing alternatives with varying degrees of value-to-cost and volatility. In keeping with the analogy, these tomatoes in region 2 are edible and could be picked now but would benefit from more time on the vine. Other tomatoes are inedible and show varying degrees of promise in blossoming into edible, ripe tomatoes before the season ends.

The true power of the option space is that it encourages activism in managing the options space. By providing an assessment of volatility, as is done with financial options, it can signal to the decision maker how much things can change over time before an investment decision must finally be made. This allows the formulation of recommendations on which alternatives could benefit the most from additional resources of time, money, or labor. In this way, the options space developed by Luehrman can be a powerful decision making tool. As Luehrman stresses, "Evaluating the project as an option space means there is more, not less, to analyze, but the framework tells us what to analyze, gives us a way to organize the effects, and offers a visual interpretation" [112]. The two key parameters in Figure 56 that define the option space are value-to-cost, or NPV_q, and volatility. The value of an option depends on a number of factors [15, 51, 91, 111]:

- *Stock Price* (S): The current price of the underlying stock or security (S)
- *Exercise Price* (X): Also known as the *Strike Price*, which the fixed price agreed upon by the buyer and seller of options as to what price the underlying security can be bought or sold, which is done by exercising the option.
- *Time to expiration* (t)
- *Risk-free rate of return* (r_f): The theoretical rate of return of an investment with no risk. In practice, no investment is without risk. Typically, a Treasury bill of a historically stable country with very little chance of credit default such as the United States is chosen as the default.
- *Variance* (σ^2): In mathematics, variance measures the spread or variability in the values taken by a random value. For options, the variance of the price of the underlying stock is measured [51]. The standard deviation is often used in place of variance to describe the spread of a distribution. The standard deviation is the positive square root of variance, or σ.

In option valuation, the *cumulative variance* ($\sigma^2 t$) is used as a measure of the total amount of uncertainty [51]. So an option expiring in two years has twice the cumulative variance as an otherwise identical option expiring in one year, given the same variance per period. Finally, by taking the square root of the cumulative variance, cumulative volatility or $\sigma\sqrt{t}$ can be used as the left axis of the option space. Including this time dimension means that how much things can change while we wait, for better of for worse, depends on how long we can afford to wait [111]. The determination of NPV_q can then be made using Equation 30 and Equation 31. Figure 57 explains the correspondences between the option variables and different aspects of a real commercial business project. Figure 58 explains the link between conventional NPV and option value, which occurs at expiration.

$$NPV_q = \frac{S}{PV(X)} \tag{30}$$

$$PV(X) = \frac{X}{(1 + r_f)^t} \tag{31}$$

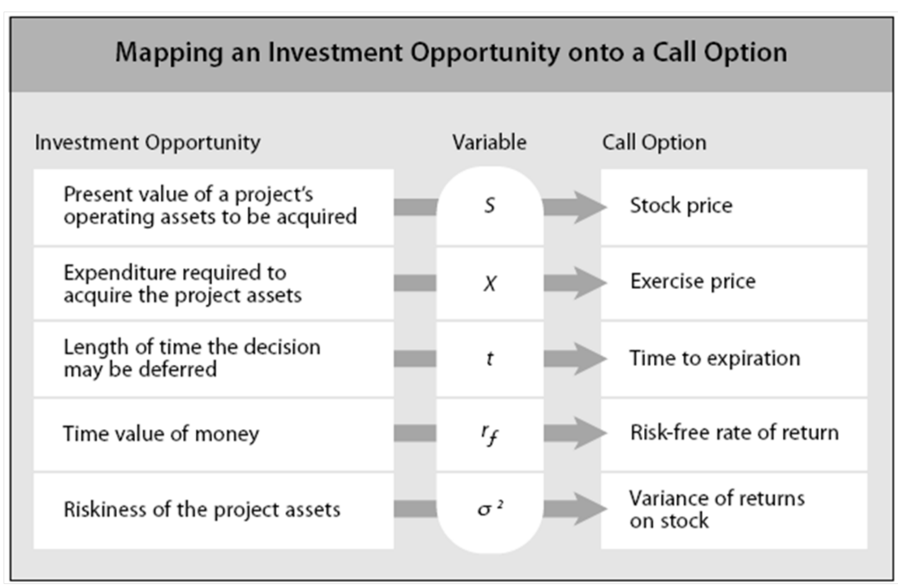


Figure 57: Mapping of Option Variables to Real Projects [111].

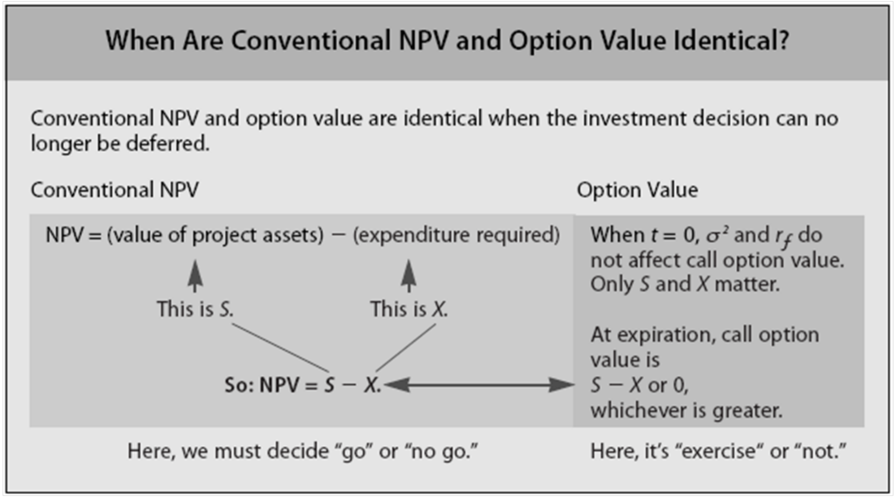


Figure 58: Relationship Between NPV and Option Value [111].

The visual framework of the option space is also visually appealing and has obvious benefits as a top-level summary for decision making when compared to those usually presented in multiattribute portfolio analysis. As Luehrman states [111]:

> By combining variables in this way, we get to work with two metrics instead of five. Not only is this easier for most of us to grasp, it also allows us to plot two-dimensional pictures, which can be helpful substitutes for equations in managers' discussions and presentations. Finally, each of the metrics has a natural business interpretation, which makes option-based analysis less opaque to non-finance executives.

Consequently, Luehrman's framework shows promise in extending the top-level summaries seen in most multiattribute portfolio displays by integrating the information necessary to distinguish between different alternatives into an additional top-level view. The primary benefit of this view is that it is capable of providing clear recommendations on which alternatives show promise for further investment based on the valuation obtained from performing the real options analysis (ROA). Hence, adapting Luehrman's method for use in acquisition decision making and AoA will result in an Architecture Real Options Complexity-Based Valuation Methodology (ARC-VM).

6.4 Developing the Acquisition Option Space

6.4.1 Mapping Option Variables to Acquisition Projects

Figure 59 depicts the mapping from a call option to acquisition metrics. The application of Real Options to military acquisition is not as straight-forward as simply describing an acquisition program in terms of a an engineering project with expected financial benefits, then using the appropriate mappings outlined by Luehrman. Nonetheless, a similar mapping can be achieved with a bit of creativity. The expense of obtaining the required capability and any needed flexibility, adaptability, etc. is now measured in terms of the complexity of the architecture. Use of the Real Options framework provides mechanisms to include uncertainty in the valuation while also discounting complexity using proper program management. These aspects are captured by the variance of the MoE and the Probability of Program Success (PoPS). The framework also helps in capturing the time value of delivering the capability sooner to the war fighter. This is reflected in the use of the acquisition time ratio.

6.4.2 Effectiveness & Time-Valued Capability

In keeping with Luehrman's approach, ARC-VM considers the effectiveness in providing the desired capability as the measurable return on the resources invested to acquire an SoS. For comparison, different MoEs must be transformed from their native units to a normalized scale ranging from 0 to 1. Various types of utility functions may be used to accomplish this, especially if it is deemed that there are nonlinear effects that must be captured as higher levels of effectiveness are reached. For example, fulfilling 60% of a capability vs. 70% may not be an additional 10% better, it might be 30% better because it is a nonlinear payoff to the warfighter. The methodology is also flexible in that separate valuations can be obtained for various MoEs or a single valuation can be made by combining the MoEs into an overall capability rating. This choice is left to the discretion of the decision makers. Care should be taken when

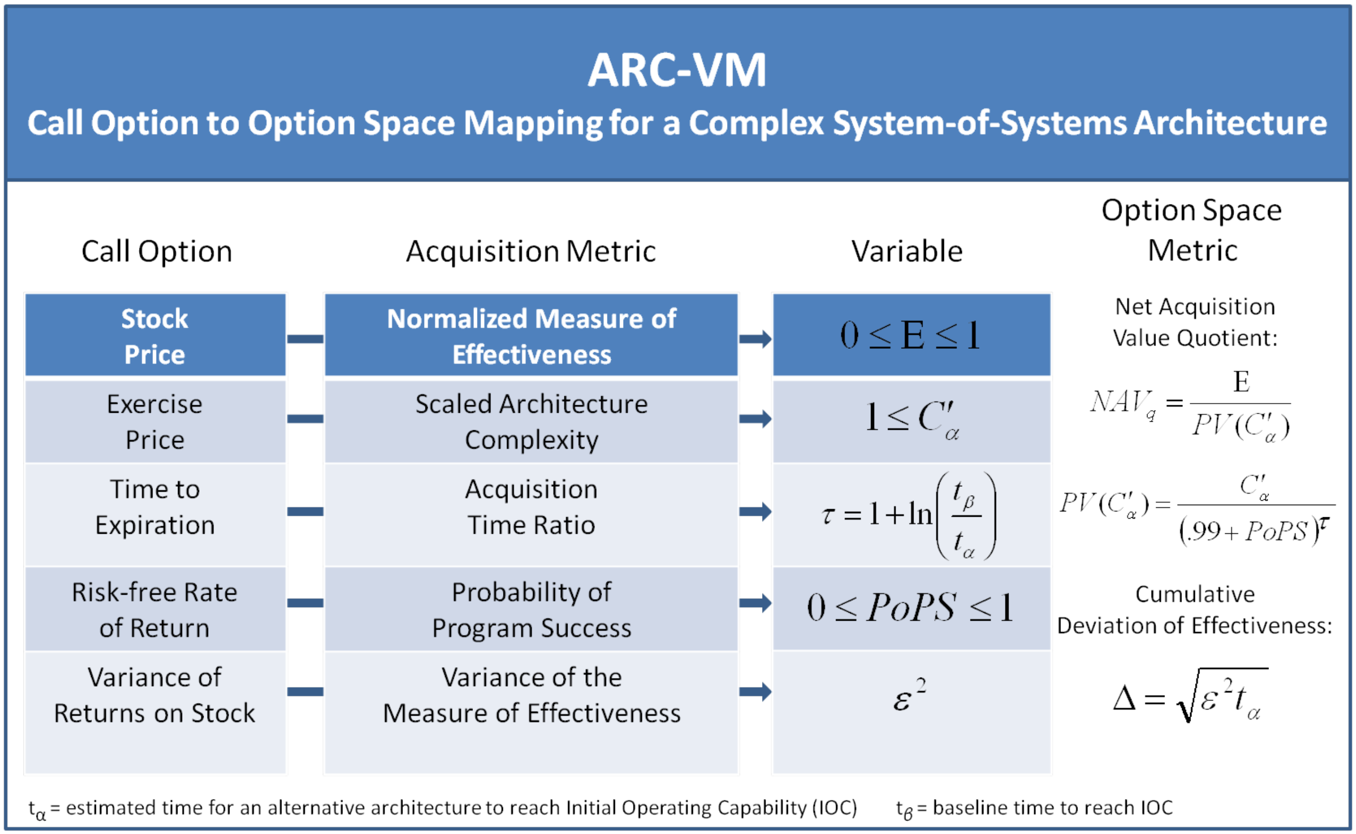


Figure 59: Mapping of Option Variables to an Acquisition Project.

combining different MoEs along with their respective variances, however. When linearly combining more than one MoE and its associated variance, if the MoEs are not independent quantities, then it may be necessary to account for any covariance, which is the strength of the dependence between the MoEs [91].

The time value of a capability was previously identified as an important consideration when evaluating alternatives. The acquisition time ratio captures this by taking the ratio of the baseline acquisition time to the estimated time to IOC for a given alternative. The baseline time is the time in which the capability must be received in order to be of value to the warfighter. The baseline time may be the same for each alternative, or may differ for each. A logarithmic function is used to transform this ratio in order to account for nonlinear effects which are assumed. The ratio has the property that an acquisition program that is delivered on time will have an acquisition time ratio of 1. A project that is projected to deliver a capability in a shorter amount of time than the baseline will have the effect of increasing the acquisition value of a project. In contrast, a project that is projected to take longer than the baseline time to deliver the intended capability will incur a penalty to the acquisition value.

6.4.3 Cumulative Variance

Luehrman states that cumulative variance is a good way to measure the uncertainty associated with business investments, but instead of using the variance of underlying project *values*, he proposes using the variance of project *returns*. This means that rather than working with the actual dollar value of the project, the percentage gained (or lost) per year is used instead. Since a project's return is determined by the project's future and present values, there is no loss of content. Luehrman goes on to state [111]:

> The probability distribution of possible values is usually quite asymmetric; value can increase greatly but cannot drop below zero. Returns, in

contrast, can be positive or negative, sometimes symmetrically positive or negative, which makes their probability distribution easier to work with.

To apply Luehrman's method to military acquisition, the MoEs of the desired capabilities now become analogous to a project's return. The MoEs obtained during M&S are typically represented as varying about some mean value, both positively and negatively. For example, the number of enemy units destroyed may be evaluated as $80\% \pm 6\%$. With this in mind, it is anticipated that using the variance of an MoE should be in keeping with Luehrman's preferred practices.

6.4.4 Risk & Probability of Program Success

For traditional business investments, the interest or discount rate is the cost of capital for the firm felt to be commensurate with the level of risk to be undertaken [134]. However, Peoples & Wilcox have observed in engineering design that "the effects of the arbitrary choice of risk-adjusted discounted rate has a large effect on the resulting design by causing design decisions to focus overly on reducing short-term development costs" [135]. For this reason, and the fact that the acquisition valuation is now based on the currency of complexity rather than dollars, the development of the AOS must rely on a surrogate metric for risk other than a financial interest rate. For an acquisition program, this estimate of risk is reflected in the PoPS. The *Risk Management Guide for DoD Acquisition* defines risk as "a measure of future uncertainties in achieving program performance goals within defined cost and schedule constraints" [61]. Therefore, risk directly impacts PoPS. Estimating the probability of an acquisition program's success is particularly useful since using probabilities to characterize and analyze uncertainty early in the design process is one method that helps bring system knowledge forward in the design process [79]. By using PoPS in this manner, the architecture complexity can be discounted by the appropriate measures used to mitigate programmatic risks. This stems from the notion that "some programs are worth the risk, and need additional resources to buy the risk down" [24]. Discounting complexity allows decision makers to make these types of trade studies. It also provides an objective way to compare acquisitions with different MoEs, complexity, and risk management profiles. For example, a highly complex architecture that has behind it a great deal of support in terms of management and resources can be compared against a less complex architecture with less available management resources.

In the context of defense acquisition, risk has three components [61]. The first is a future root cause. The second is a likelihood assessed at the present time of that future root cause occurring. The third is the consequence of that future occurrence. The implementation of methods for risk identification, management, and mitigation are also well documented by the DoD and among the different branches of the armed forces [61]. The Army, Navy, and Air Force each have developed PoPS tools for identifying and tracking different sources of acquisition program risk throughout the acquisition life cycle [9, 23, 66, 154]. The following is a brief summary of some of the programmatic risk factors that should be taken into account when analyzing programmatic risk [125, 135, 144]:

1. *Technical*: Exposure to the chance that development of critical technologies will not meet program objectives within cost and schedule constraints. In assessing technical risk, program managers must address the uncertainty in their estimates about how much time and effort it will take to make new technologies work. One purpose of evolutionary acquisition is to provide time to better manage technology risk and avoid adverse cost and schedule outcomes that often result from trying to achieve difficult requirements in one step. Proxy measures such as Technology Readiness Levels (TRLs) can be used in determining the technical risk factor.

2. *System Integration*: Exposure to the chance that new and existing technologies or systems being employed in the SoS architecture may not work together and/or interact with operators and maintainers to meet program objectives within cost and schedule constraints. When developing capabilities that are part of a larger war fighting capability, individually the component programs might appear to be a low or moderate risk, but in combination with other programs, the overall risk might be much higher due to coordination and integration issues. A classic example occurred during the Grenada invasion when Army and Navy communications systems did not interact well during the joint operation.

3. *Design Risk*: Exposure to the chance that the SoS will not result in effective operation or be easy to produce. Decisions made early in the design process quickly establish not only the performance but also the ease of manufacture. Design complexity has also increased with the availability of more sophisticated design tools such as electronic product models and computational techniques (e.g. finite element analysis).

4. *Production Risk*: Exposure to the chance that the facility, labor, manufacturing processes, and procedures will fail to produce the weapon system within the time and cost constraints. Producibility—or "production capability"—is a function of the design, production facilities; management skills, processes, and experience; and workforce skills and experience. As alternatives grow more complex, the challenges in producing the given SoS are expected to grow.

5. *Business Risk*: Exposure to the chance that the overall acquisition strategy for a program will not result in the desired cost, schedule, and/or performance outcomes. An example would be the risks inherent in utilizing an entirely new business model to acquire a system. Decisions about the process to select who

will build elements of the SoS, the standards to which it will be built, and the schedules for designing and building them all entail risk that is not always appreciated up front. Evaluation of the business risk entails assessing (1) the extent to which the acquisition strategy can result in selection of the most effective, efficient design and most effective, efficient production entities; (2) whether cost estimates and schedules are valid; (3) whether proper government oversight organizations are in place; and (4) whether project personnel with proper training and experience are available. Evaluation of business risk should also take into account financial uncertainty, as general economic conditions often have a significant impact on acquisition program funding. Financial uncertainties can also be related to nonrecurring or recurring costs, price, or demand.

Since a SoS usually has stakeholders of varying influence, each with their own constraints and requirements for their respective systems, the organizational complexity of the acquisition effort should influence the determination of each separate risk factor as well, or be included as an additional risk factor [72, 145]. A detailed analysis of the complexity of the various programs/organizations involved in the acquisition effort by analyzing their collective interrelations and interdependencies would be of great benefit to the decision making process and is recommended for future work. Referring back to Figure 3, organizational risk is defined in a similar manner as well:

- *Organizational Risk*: Exposure to the chance that the number, diversity, and competing interests and priorities of separate stakeholders will not result in the desired cost, schedule, and/or performance outcomes. Evaluation of the organizational risk entails assessing whether or not all of the stakeholders who may have a vested interest in the acquisition outcome have been properly identified, the level of involvement of each stakeholder, the amount of influence stakeholders are able to exert, and the management process in place to facilitate effective communication, address concerns, and increase knowledge and awareness of constraints and developmental changes [69, 145].

Together, these factors describe the overall risk profile for an acquisition program. These risk factors interact in numerous ways to affect the cost, schedule, and performance of developing a particular alternative. Technology Readiness Levels have been previously defined and extensively used. A summary from Ref. [125] is provided as Figure 60. Following the template of defining TRLs, Murphy researches readiness levels for system integration and engineering and manufacturing readiness. These are presented in Figure 61. Integration Readiness Levels (IRLs) developed by Sauser are included in Murphy's research [144]. Murphy then proposes readiness levels for business processes and design process levels, which are presented in Figure 62. In similar fashion, a rating scale following the same format as the one used to capture technical, system integration, design, production, and business risk could be developed to capture organizational risk factors as well. Overall, by considering all six factors, a detailed assessment of PoPS can be obtained.

Technology Readiness Levels
1. Basic principles observed and reported
2. Technology concept and/or application formulated
3. Analytical and experimental critical function and/or characteristic proof of concept
4. Component and/or breadboard validation in laboratory environment
5. Component and/or breadboard validation in relevant environment
6. System/subsystem model or prototype demonstration in a relevant environment
7. System prototype demonstration in an operational environment
8. Actual system completed and qualified through test and demonstration
9. Actual system proven through successful mission operations

SOURCE: DoD, May 2005.

Figure 60: Technology Readiness Levels [125].

Integration Readiness Levels
1. An **interface** between technologies is identified with sufficient detail to allow characterization of the relationship.
2. There is some level of specificity to characterize the **interaction** (i.e., ability to influence) between technologies through their interface.
3. There is **compatibility** (i.e., a common language) between technologies to orderly and efficiently integrate and interact.
4. There is sufficient detail in the **quality and assurance** of the integration between technologies.
5. There is sufficient **control** between technologies necessary to establish, manage, and terminate the integration.
6. The integrating technologies can **accept, translate, and structure information** for their intended application.
7. The integration of technologies is **verified and validated** with sufficient detail to be actionable.
8. Actual integration is completed and **mission qualified** through test and demonstration in the system environment.
9. Integration is **mission proven** through successful mission operations.

SOURCE: Sauser et al., 2008.

Engineering and Manufacturing Readiness Levels
1. System, component, or item validation in laboratory environment or initial relevant engineering application or breadboard, brass-board development.
2. System or components in prototype demonstration beyond breadboard, brass-board development.
3. System, component, or item in advanced development. Ready for low-rate initial production.
4. Similar system, component, or item previously produced or in production. System, component, or item in low-rate initial production. Ready for full-rate production.
5. Identical system, component, or item previously produced or in production. System, component, or item in full-rate production.

SOURCE: DoD. 2005.

Figure 61: System Integration & Engineering and Manufacturing Readiness Levels [125].

Business Processes

1. Using a new, unproven approach to source selection. Encouraging new sources of supply. Acquiring new technologies without well-established cost-estimating relationships. Requiring new government and/or contractor organizations to be formed.
2. Using new procurement process in established industry. Cost-estimating relationships only at high levels. Requires expansion of government and contractor organizations.
3. Evolutionary change from prior acquisition strategies. Good cost-estimating relationships. Existing government and contractor organizations can easily adapt to changes.
4. Using same approach to buying similar products. Well-established cost-estimating relationships exist. Experienced government and contractor organizations involved.
5. Acquiring more of what has been successfully bought before. Using the same contractor and government organizations.

Design Processes

1. New, unproven processes. New design tools under development. New design organization.
2. Large expansion of existing design organization. Many new designers and supervisors unfamiliar with design tools and processes.
3. Existing design organization using radically changed design tools, processes, and/or technologies.
4. Experienced design organization using new design tools with proven processes.
5. Experienced design organization using existing, proven design tools and processes.

Figure 62: System Integration & Engineering and Manufacturing Readiness Levels [125].

6.4.5 Scaling & Discounting Architectural Complexity

To recap, an alternative is deemed desirable when $NAV_q \geq 1.0$ and a poor investment when $NAV_q < 1.0$. Depending upon the actual formulation of C_α obtained after selecting architecture complexity sub-measures, the range of scores calculated for different architectures can encompass a wide range. This was demonstrated previously in Figures 52 & 53. The end result is that C_α must be scaled prior to use in the Real Options calculations. The scaling should be carefully performed so that consistency of NAV_q values is maintained as parameters such as effectiveness, PoPS, or the time required to develop the architecture vary. To accomplish this, a lower bound of 1 is chosen so that an alternative with 100% risk is deemed a poor investment no matter the capability performance rating or the time of delivery. This means that the maximum value of E, which is $0 \leq E \leq 1$, is restricted to the minimum value of scaled architecture complexity, because $1 \leq C'_\alpha$. This makes the ARC-VM approach different from many other traditional option valuations where the underlying stock price can exceed the strike price of the corresponding option. With this in mind, the following criteria should be adhered to when defining the scaling function:

- Continuous, with $1 \leq X$
- Monotonically increasing so that rank order is preserved *i.e.,* for all X & Y such that X $\leq$ Y, then f(X) $\leq$ f(Y)

Different options exist for defining the necessary scaling function. Based on the lessons learned from measurement theory and utility theory, both linear and log-linear transformations will be discussed. First, in order to define a line two points are needed. The first point is known, since the y-intercept occurs at a value of 1, corresponding to the minimum allowed value of scaled architecture complexity. Determination of the second point requires developing an additional parameter. Thus, when determining the choice of linear scaling function it is necessary to define a baseline acquisition. This is depicted in Figure 63.

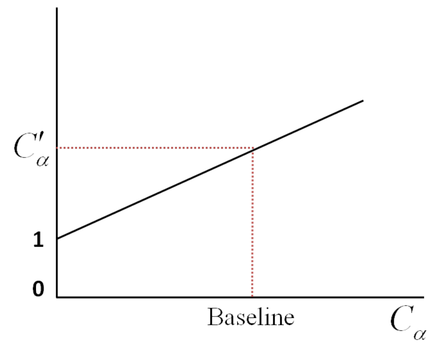


Figure 63: Linear Architecture Complexity Scaling Function.

The baseline architecture should represent a minimum acceptable alternative, such that $NAVq = 1.0$. A guideline for setting the baseline is to assume an acquisition program that is delivered on time, so that $\tau = 1$. Also, assume a 66% PoPS. This guideline is derived from the U.S. Navy's PoPS guidance for naval acquisition programs [66]. The Naval PoPS guidebook states that an acquistion program with a PoPS in the range of 66%–90% is a program that has "identified some significant issues with providing capability, supportability, and/or life cycle SE requirements within approved cost and schedule constraints, but mitigation strategies are being executed" [23, 66]. In fact, a PoPS utility curve can be defined based on guidance like that provided in the PoPS guidebooks for the different branches of the armed services. For example, programs with PoPS less than 66% can be scored as having zero utility, while programs above the 66% threshold would have increasing utility. This PoPS utility function can be used directly in the ARC-VM real options formulation if the utility scores are on a normalized scale between 0 and 1.

Next, a baseline normalized MoE of $E = 0.75$ is assumed. This estimate may vary based on the specific capability being addressed and the specific requirements set for the SoS. When these are defined, a C'_α of 1.24 is the result. This scaled architecture complexity score can be paired with a raw architecture value score to define the baseline C_α. Curves can then be fit between these two points. Since it is not likely that there will be prior knowledge about which ranges of raw complexity scores correspond to different levels of acquisition difficulty, some recommendations are given. An alternative that is or close to representing an agreed upon baseline acquisition program can be selected, and its C_α score used as the baseline value to pair with the 1.24 value previously calculated for C'_α. Another method is to calculate the mean C_α of the sample and use that to define the second set of coordinates. Along with this, the standard deviation of the raw complexity scores can be used such that alternative with the highest calculated raw complexity (largest value of C_α) is never more than 1-2 standard deviations away from the baseline. There are obvious drawbacks to using the linear scaling method, chief among them being that the need to specify a baseline creates a degree of subjectivity that can potentially limit the usefulness of the overall analysis.

The second option for scaling the architecture complexity scores is to perform transformations using logarithmic functions. Logarithmic transformations are chosen since it is desirable to maintain the relative magnitudes of the comparative ratios between alternatives [48]. Along these lines, the following general form can be used:

$$Y = a + k_c \log_b(X) \tag{32}$$

where X corresponds to a raw architectural complexity score, C_α, and Y is the desired scaled architecture complexity score, C'_α. When $X = 1$, Y is equal to a since $\log_b(1) = 0$ Therefore $a = 1$, leaving us with the task of defining the constant k_c. The only problem that we may encounter is that when $X < 1$, negative values appear for $\log(X)$, resulting in C'_α scores that approach $-\inf$. This is easily corrected for by scaling the raw architecture complexity scores by an appropriate constant (which is allowed since the multiplicative form of the utility function is used) so that the minimum complexity score that can be achieved by the simplest valid architecture is set equal to one. As a reminder, architectures with zero complexity scores violate the definition of a SoS in some way, so when an undefined $NAVq$ results, the conceptual framework still holds. The remaining step is to determine the constant k_c.

Determining the constant c requires elaborating on a few important concepts with regards to architectures. First, a fundamental assumption of this research is that a capability (or set of capabilities) can be decomposed hierarchically from capability to high-level activities down to individual tasks or functions. The decomposition, or functional analysis/allocation, should be performed in such a manner such that functionality can be allocated properly to individual systems. It is critical that the functional decomposition is performed correctly, because this sets the level of abstraction at which the analysis will be conducted. With that said, it is possible that a function in the context of one capability may be considered a higher-level activity in another and vice versa, depending upon the level of abstraction and the definition of what boundaries constitute a system. Thus, a functional decomposition of different capabilities done could yield different numbers of functions to be distributed within the architecture, directly impacting the architecture complexity measure. A method of scaling architecture complexity should be robust in dealing with this issue, so that the scaling function can be used to analyze architectures that deal with a wide variety of capabilities in numerous different contexts. Accordingly, k_c can be defined as such:

$$k_c = \frac{1}{b \log_b(F)} \tag{33}$$

where F is the total number of functions that help define the architecture and level of abstraction. This form of k_c allows normalization by the number of functions that define the architecture, taking into account the base (b) of the logarithm being used. Because the normalized MoE (E) is on a 0–1 scale, the base 10 logarithm should be used, making $b = 10$ the preferred choice. If the natural logarithm were to be used, to maintain the proper scaling relationship, E should be placed on a 0–exp scale, where exp is the exponential (also commonly referred to as just e). Defining k_c in this manner leads to the final form of the architecture complexity scaling function:

$$C'_\alpha(X) = 1 + \left(\frac{1}{10\log_{10}(F)}\right)\log_{10}(X) \tag{34}$$

To illustrate the normalization that occurs using this function, when $F = X$ a value of 1.1 is the result. As an added note, because E is defined on a 0–1 scale, relatively small changes in scaled architecture complexity values can have a big impact on the resulting $NAVq$. In the case of the linear transformation, this makes it very important to define the slope of the linear transformation properly in order to avoid over inflating or under inflating the scaled architecture complexity scores. Providing a logarithmic transformation helps alleviate this concern, making this method the preferred choice for use in the Real Options valuation framework.

6.4.6 The Acquisition Option Space

Once the mapping from a call option to acquisition metrics is completed, an options space similar to that put forth by Luehrman [111, 112] but more suitable for acquisition decision making can be defined. This options space is referred to as the Acquisition Option Space (AOS). The AOS allows the visualization and evaluation of alternative architectures of varying complexity, capability, and risk. An example AOS is shown in Figure 64.

The AOS is inverted with respect to Luehrman's option space, primarily for ease of plotting. The two key parameters that define the AOS are the Net Acquisition Value (NAV) Quotient and the cumulative deviation of effectiveness. These are symbolized by ($NAVq$) and Δ, respectively. An alternative with a $NAVq \geq 1.0$ is desirable,

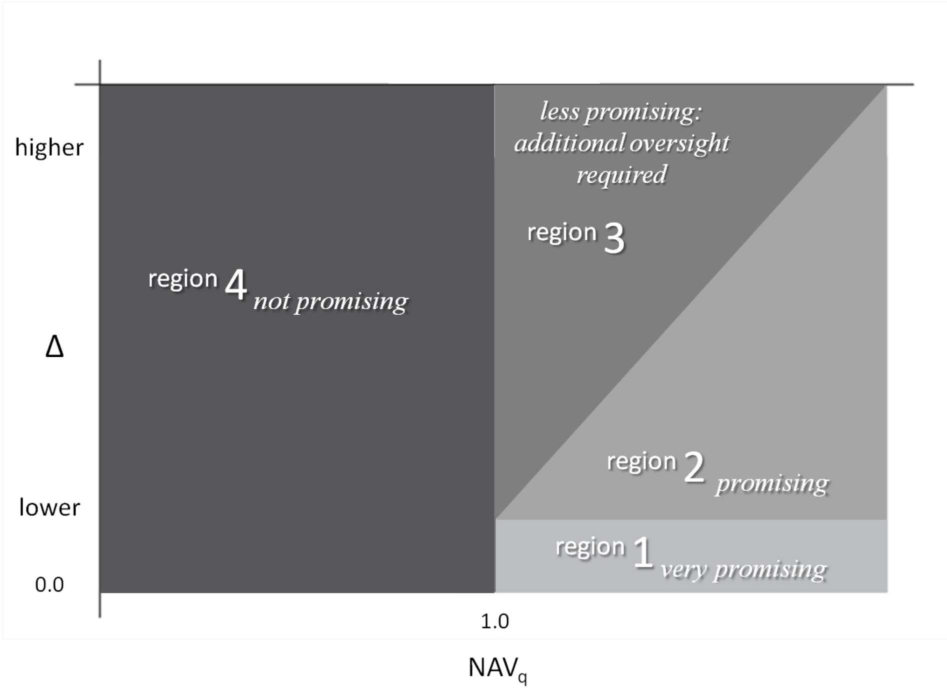


Figure 64: The Defense Acquisition Option Space.

whereas an alternative with a $NAVq < 1.0$ is not. $NAVq$ provides a balanced assessment of effectiveness, complexity, risk, and time-valued capability while Δ is a measure of the uncertainty that results from attempting to create complex architectures that must function in a dynamic, shifting threat environment. Δ is comprised of two components. The first component of Δ, ϵ, assumes that SoS effectiveness can be captured as a distribution. Thus, M&S results should provide the estimated deviation in effectiveness that results under different operational circumstances. The second component, t_α, compounds this deviation by the time it takes to deliver the capability to the warfighter. As the time to field a capability increases, so does the chance that changes in the threat environment, the budget landscape, or possibly shifting requirements will impose cost and schedule growth, jeopardizing the acquisition program. Additionally, in this time the planned effectiveness of the architecture under development may decrease or even become obsolete as an enemy adapts and evolves, or other technologies or countermeasures are introduced. On the other hand, providing additional time for alternative to mature may raise the chance that critical breakthroughs are achieved. In reality, there are any number of unforeseen circumstances which may arise.

The AOS is not meant to replace the traditional scatter plot and matrix of alternative comparison views previously presented in Figure18 and Figure19, but is rather meant to extend these views and allow for more informed decision making. The AOS accomplishes this by integrating the information necessary to provide a balanced valuation of alternative architectures while providing clear recommendations on which alternatives show promise for further investment. The latter is accomplished by subdividing the AOS into 4 distinct regions. Alternatives that occupy the first region in the upper right-hand corner of the AOS are high value alternatives with low Δ. These represent alternatives that have obvious benefit and that can be pursued immediately with great confidence. At the other extreme are alternatives with $NAVq < 1.0$ and thus occupy the left-hand side of the AOS. These alternatives show little promise since the complexity costs are too great in comparison and will provide little or no value when considering the amount of risk and resources necessary to bring them to fruition. Region 1 & region 4 alternatives are at the extremes and are easy decisions to make. The regions in between are where architectures of varying prospects are and that require closer scrutiny. The decision maker may then decide whether the level of effectiveness provided by an architecture in either regions 2 or 3 would be worth an additional investment of resources. As Δ grows larger and an alternative moves further down into region 3, the acquisition becomes much less promising in providing a satisfactory return on the resources invested. Early termination of these programs is recommended if significant design challenges are encountered early on. For these alternatives, as well as those that occupy region 4, a wait and see approach or further investments in related R&D efforts are most likely the best courses of action.

The AOS is not a static representation. Time is a crucial element. As the acquisition program matures and time to IOC draws near, schedules change and the alternatives in the option space will move nearer to the x-axis as cumulative volatility decreases. The alternatives will also move sideways depending upon changes in PoPS, schedule, and changing estimates of effectiveness. Ideally, all the alternatives will inch closer to Region 1, with the most promising alternatives approaching Region 1 the quickest. Also, the lines that define the boundaries between the different regions potentially contain important information as well. Luehrman shows this by devising an alternative method, which can be seen in Figure 65.

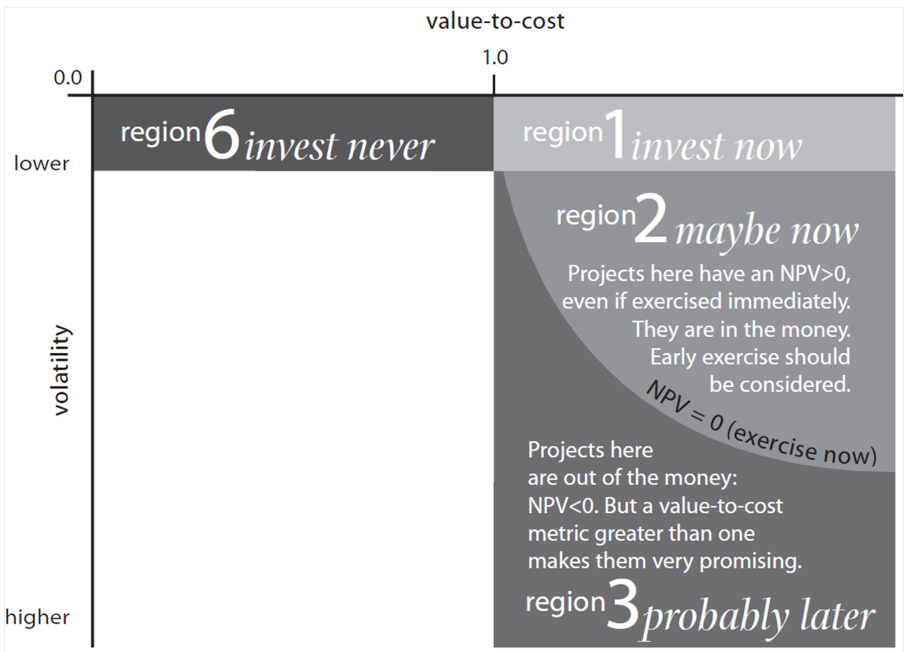


Figure 65: Dividing the Option Space Using Curved Regions [112].

Luehrman's approach to drawing the curved lines in Figure 65 makes use of the Black-Scholes equation in dividing the option space. This is discussed in Figure 65, where the curve is derived by holding the r_f and σ constant in the Black-Scholes equation as t varies, then solving for the value of the value-to-cost metric that corresponds to $NPV = 0$. In the extreme case of $r_f = 0$, Luehrman notes, the curve is instead a vertical line corresponding to points where $NPV = 1$. As r_f increases, the slope of the curve decreases, causing the curve to bend to the right [112]. Use of the Black-Scholes equation, with its many assumptions, is not applicable to the AOS. However, there are guidelines that can be developed to aid decision makers. For example, if some basic baseline parameters can be identified, the impact on the AOS becomes readily apparent. Acquisition professionals typically work with baseline values in order to make meaningful comparisons between alternatives. The baseline values that provide the most benefit in this context are baseline values for the normalized measure of effectiveness as well as a baseline value for the variance of the measure of effectiveness. These two values can be represented by E_B and ϵ_B, respectively. A third value, $t_m in$, will prove useful as well, as this represents a minimum funding period for acquisition programs. For multi-year programs, this basic unit may be a year. For programs of shorter length, quarters or weeks can be used. With these parameters defined, the location of the vertical boundary between the first and second AOS regions can be set using ϵ_B as a guide. ϵ_B can be multiplied by $\sqrt{t_{min}}$. So if t_{min} is one, then the horizontal boundary is equal to ϵ_B. Figure 66 illustrates the effects of changes in ϵ_B on the AOS boundaries.

When a capability must be performed within very tight specifications, as measured by minimum variance, then the area of region 1 decreases. This makes it more difficult for alternatives to reach region 1 as they traverse the option space over time, since they must have an extremely low cumulative deviation of effectiveness. In contrast, as requirements are relaxed and the desired capability level can take a wider range of values, say $80\% \pm 10\%$ vs. $80\% \pm 5\%$, the area of region 1 expands. As can be seen in the figure, the boundary defining regions 2 and 3 shift as well. In order to ascertain the effects of changes in E_B, first it is necessary to specify how the boundary line between regions 2 and 3 can be defined. In order to specify a line two points are needed. The first point is given. This is the point of intersection between regions 1-3. The y-coordinate of the second point can be defined in terms of baseline parameters,

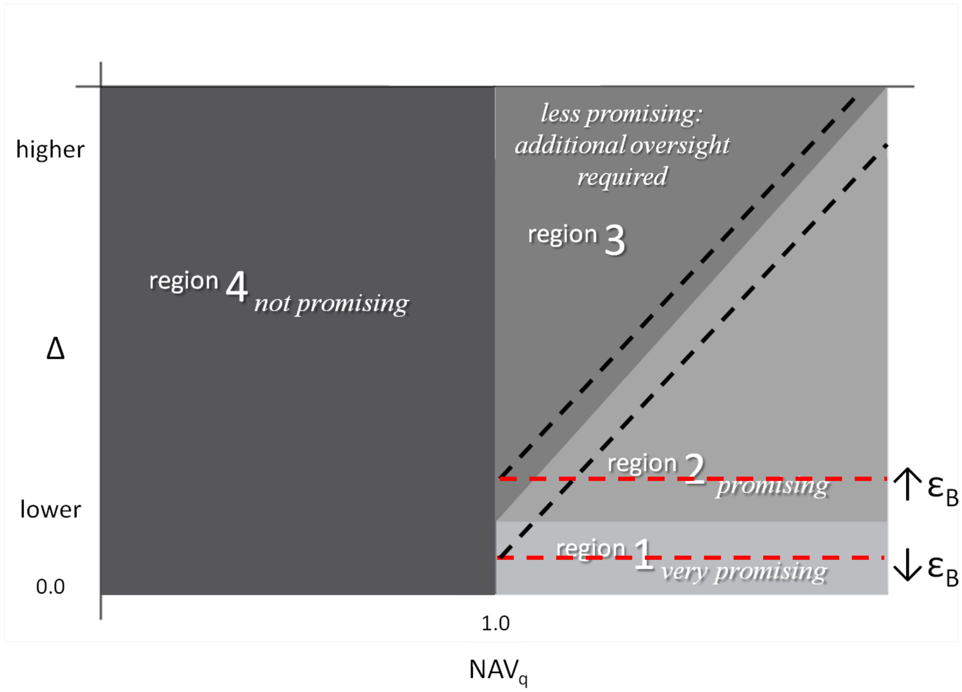


Figure 66: The Effect of Baseline Variance of Effectiveness Requirements on the Acquisiton Option Space.

and will be referred to as Δ_B. The following equation defines this variable:

$$\Delta_B = \epsilon_B \sqrt{t_\beta} \tag{35}$$

Thus, Δ_B is the cumulative deviation of effectiveness that an alternative with baseline variance and that can be developed within the desired time frame would have within the AOS. This leaves the remaining x-coordinate that still must be determined. This requires determining the $NAVq$ for the baseline alternative. Since t_β is used to define Δ_B, this means that $t_\alpha = t_\beta$ can be substituted into the $NAVq$ equation. This results in an acquisition time ratio of 1. If we also assume an alternative of minimum complexity and maximum PoPS, then the following equation results:

$$NAVq = E_B(.99 + 1) \tag{36}$$

This $NAVq$ defines the x-coordinate so that the boundary line between regions 1 and 2 can be drawn. Using Equation (36), it is possible to see the effect of changing E_B. Figure 67 provides a visual example. Specifying that greater effectiveness is required means increasing E_B. As E_B increases, $NAVq$ value used as the second x-coordinate for defining the boundary increases as well. This causes the slope of the boundary line to decrease. The net effect is that the area of region 2 decreases while the area of region 3 increases. This corresponds with the intuitive notion that more stringent requirements will make it harder for alternatives to be rated as promising. If the opposite occurs, and baseline effectiveness requirements are relaxed, the $NAVq$ coordinate decreases and the slope of the boundary line increases. Thus region 3 increases in area, meaning that more alternatives will be able to meet the requirements and be recommended as promising. At approximately $E_B = 50\%$, region 3 vanishes alltogether and is replaced entirely by region 2. Past this point, the region 4 area begins to diminish as well.

Lastly, it is possible to incorporate the effects of a decision maker's risk tolerance into the AOS analysis. The contributions by von Neumann and Morgenstern highlight the importance of accounting for a decision maker's preferences or indifferences [92]. Therefore, utility theory can be used to described individuals or groups who are risk neutral, or who are risk provers and take risks, or who are risk averse and wish to avoid risks. Figure 68 shows how this can be incorporated into the AOS, whereas a decision maker that is risk proverse will tend to inflate the area of region 2, while a decision maker that is risk averse will tend to deflate the area of region 2. The following equations detail one possible method for defining the risk proverse/averse curves using the previously defined baseline variables:

$$NAV_q^{Max} = 1 \times (1 + .99)^{\left(1+\ln\left(t_\beta/t_{min}\right)\right)} \tag{37}$$

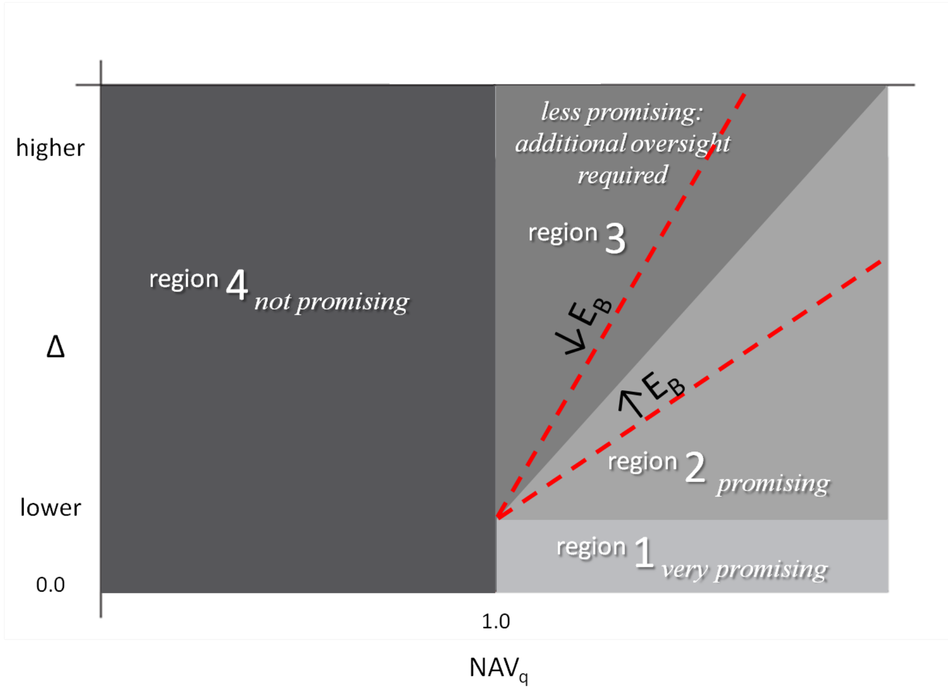


Figure 67: The Effect of Baseline Effectiveness Requirements on the Acquisiton Option Space.

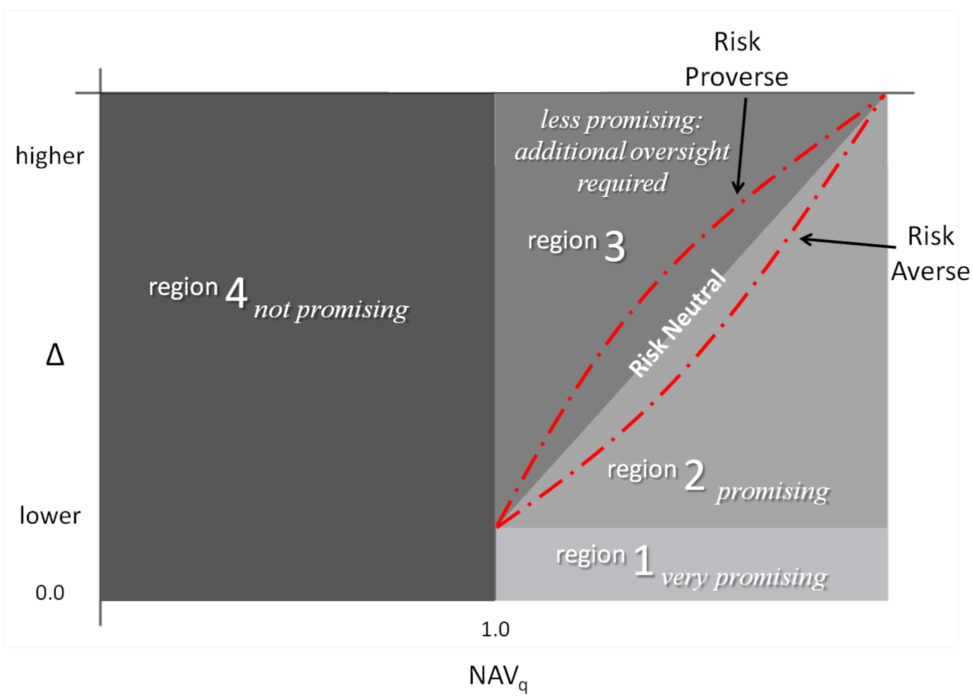


Figure 68: The Effect of Risk Tolerance on the Acquisition Option Space.

$$f = \left(\frac{NAVq - 1}{NAV_q^{Max} - 1} \right) \tag{38}$$

as $NAVq$ varies from 1 to NAV_q^{Max}. Then:

$$g = (1 - f) \tag{39}$$

$$\Delta_N = m(NAVq) + b \tag{40}$$

where m and b are the slope and y-intercept of the risk neutral boundary line between regions 1 and 2. This leads to the calculation of the risk adjusted cumulative deviation (Δ_R) for a given $NAVq$ so that the risk proverse/averse curves can be drawn:

$$\Delta_R = \psi_R(f \log_2 f + g \log_2 g)\Delta_B + \Delta_N \tag{41}$$

where ψ_R is the risk tolerance factor and $-1 \leq \psi_R \leq \psi_R$. For a decision maker with a strong risk aversion, $\psi_R = -1$. Conversely, a decision maker that is very risk proverse will be characterized by a $\psi_R = 1$.

In summary, the adaptation of Luehrman's option space provides a useful conceptual and visual framework for depicting analysis results, and serves to augment the traditional AoA and visualizations in current practice today. Though application of Luehrman's framework is not straightforward, this research details at least one possible way in which this can be accomplished. It should be emphasized that the objective of applying Luehrman's methodology is not to produce a specific option price. To develop a single option value for each alternative would require the use of established option pricing techniques with their many underlying assumptions. These assumptions may not be readily applicable to the acquisition of complex military SoS. Rather, the principal advantage gained in developing the AOS is that it provides an analytical framework for the formulation of recommendations during an AoA. More specifically, it provides a detailed analysis on which alternatives could benefit the most from additional resources of time, money, and/or manpower. Furthermore, using an objective measure of architecture complexity as the basis for determining valuation should help mitigate the effects of overly optimistic estimates of schedule, effectiveness, and risk. Decision makers will be able to decide more clearly if such estimates are warranted for an alternative of given complexity. The inclusion of uncertainty using the cumulative deviation of effectiveness as one of the primary axes of the AOS should also serve in mitigating these effects. Overall, the AOS provides a rigorous, repeatable, and traceable manner in which to make acquisition decisions, supporting the development of defensible recommendations. The AOS is a powerful decision aid, structured in such a way that it provides understandable interpretations using an intuitive visualization format. This directly supports the stated research objective by ensuring that the advantages and disadvantages of each alternative are presented in a clear and unbiased manner.

6.5 High Level ARC-VM Summary & Overview

With the Real Options framework in place, it is now possible to fully describe the entire ARC-VM methodology. The methodology was created with the intention of meeting the overarching goal of aiding decision makers in the down-selection of alternatives for the implementation of a cost-effective, evolutionary acquisition strategy.

The steps in applying the methodology are summarized in Figure 69. ARC-VM begins with a clear definition of the capability requirements and their relationship to one another. This step also includes defining the functional relationships between tasks and activities that comprise the needed capability or capabilities. Next, candidate systems must be identified that can fulfill the required tasking, resulting in a mapping of the system-to-task functional allocations for each system within the SoS. This determines the possible system portfolios that will be used later to generate alternative architectures. The next step is to determine the needlines that exist between systems. This will aid the system architect in evaluating the actual vs. desired IOLs that exists between systems. This also helps to determine the amount, type, and make-up of the resources that need to be shared between systems to fulfill the needed capability. Once this is complete, alternative system portfolios must be identified. The portfolios define which systems will be grouped together as a SoS, making it possible to move on the next step of generating feasible alternative architectures that can vary in force structure, patterns of collaboration, IOL, process sequencing, and technology.

Once these alternative architectures are generated, assessing the performance of each through M&S becomes necessary. Since a complex SoS architecture varies in many aspects, it is important to assess the appropriate M&S environments that will capture the relevant features of SoS performance. Along with this, cost estimates can be generated using appropriate cost models. Down-selection to a smaller number of alternatives can also be accomplished at this point in time, if desired. At this stage, with the alternatives fully described, it becomes possible to calculate the architecture complexity score for each alternative. With the architecture complexity score in hand, the next step is to perform the AoA by developing the AOS, the Cost-Effectiveness scatterplot, and any other top-level portfolio analysis summaries that are needed. The process need not stop here, as further down-selection may result in alternatives that can be more carefully scrutinized with higher-fidelity analysis tools

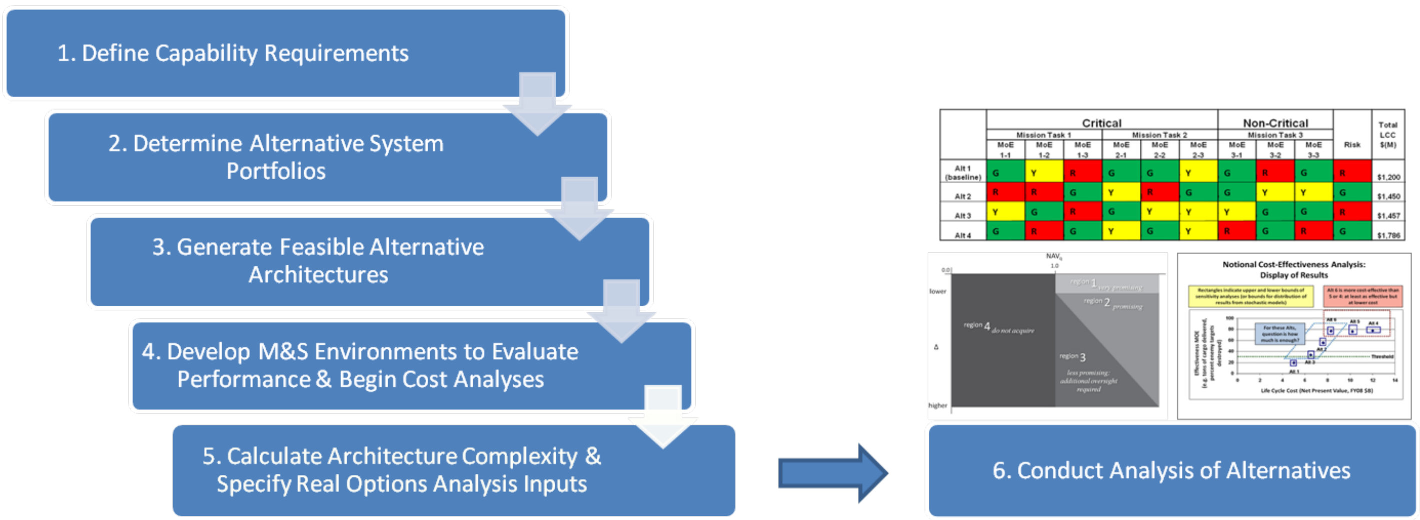


Figure 69: Overview of ARC-VM Methodology.

and techniques. The ARC-VM methodology is meant to support iterative design efforts. With the steps of the methodology clearly enumerated, Figure 70 serves as a visual overview of the progress up to this point. Now that ARC-VM has been fully developed and described, the remaining task is to demonstrate that the methodology meets the aforementioned research objectives. This can be best accomplished by demonstrating implementation of the method using an appropriate, fully described mission scenario. This requires selecting an appropriate architecture for a military SoS so that the needed capability can be delivered to the warfighter in a timely and ongoing fashion. To gauge the effectiveness of the military SoS in meeting those capability requirements during the early stages of acquisition, M&S will be used. Creation of a M&S environment will allow meaningful comparisons to be made. Specifically, three critical questions must be answered when dealing with military SoS architectures. These represent key aspects of a SoS architecture that affect SoS cost, schedule, and performance that need to be well understood.

1. Which systems should be included in the SoS? *i.e.,* What is the system portfolio?
2. How do these systems work together? *i.e.,* What is the pattern of interoperability & collaboration present in the architecture?
3. How does this architecture respond to changes in force structure?

A Suppression of Enemy Air Defenses (SEAD) capability provides the desired level of complexity for a military SoS acquisition test case for a number of reasons [71]. SEAD requires that a number of diverse, independent, geographically distributed, and networked assets interoperate to provide the needed capability. Also, alternative SEAD architectures can include both legacy and to-be-designed systems as well as the use of new technologies. SEAD is defined as any activity that neutralizes destroys, or temporarily degrades enemy surface-based air defenses by destructive and/or disruptive means [96]. Systems such as Surface-to-Air Missile (SAM) sites, Anti-Aircraft

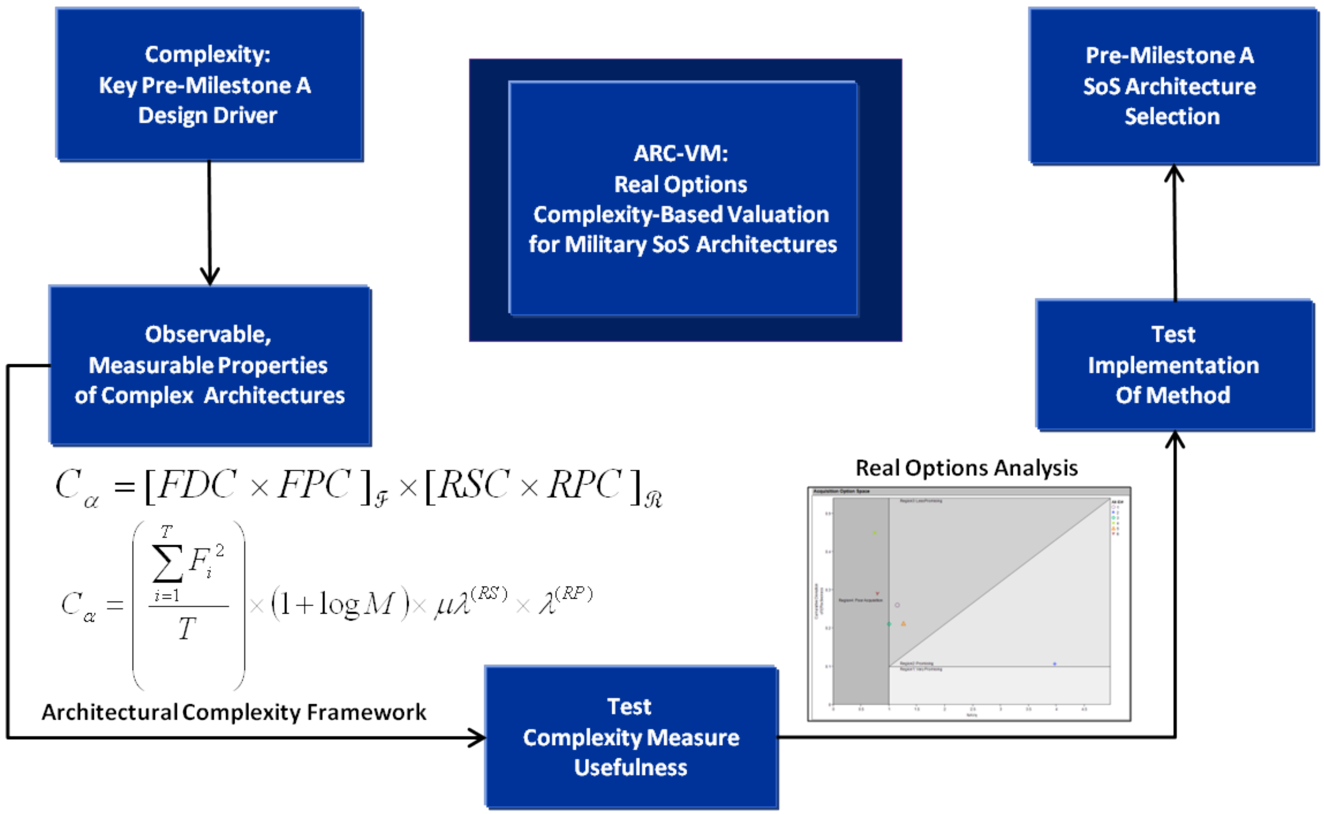


Figure 70: Overview of Research Steps.

Artillery (AAA), early warning (EWR) and fire control radars, and Ground Control Intercept (GCI) sites can be combined by potential adversaries into an Integrated Air Defense System (IADS). Over time, IADS have become increasingly complex and can differ widely in terms of organization, sophistication, and operational procedures. The widespread proliferation of weapon systems and continual improvements in their speed, range, accuracy, stealth, and lethality require joint forces to be more responsive, flexible, and integrated. Since SEAD can be conducted jointly, a multitude of various system types can be included within the architecture. These range from sea, land, air, and space-based assets to manned and unmanned systems. Also, sophisticated communication systems are needed to enable and enhance Command and Control since enemy air defenses can be mobile or stationary and pose a significant threat to current military assets. For purposes of developing the complexity measure, a SEAD scenario that focuses on Area of Responsibility (AOR)/Joint Operating Area (JOA)-Wide Air Defense System Suppression provides the desired complexity for a military SoS architecture test case. This means that SEAD is conducted against specific enemy air defense systems throughout the AOR/JOA to degrade or destroy their major capabilities/effectiveness. The duration and level of disruption depends upon the mission objectives and the sophistication of the IADS [96].

Demonstration of the validity of the ARC-VM method will be achieved by further developing the SEAD AOR/JOA example problem into an appropriate mission scenario. A scenario-based approach is selected since the storyboard nature provides a useful context for application of the research methodology [157]. Scenarios are an integral part of an AoA, and define the operational locations, the enemy order of battle, and the corresponding enemy strategies and tactics. To be useful, the SEAD AOR/JOA scenario is developed with consideration of the capability need, applicable constraints and assumptions, and the physical environments expected Ref. [127]. For security classification purposes, it should be noted that all data and parameters used in this analysis is either derived from open, unclassified sources or purely from the author's estimation. Development of this mission scenario is only intended to provide a clearer understanding of the methodology and its application.

CHAPTER VII

VALUATION OF SEAD ARCHITECTURES USING ARC-VM

7.1 Step 1: Define Capability Requirements

An integral step in defining how a capability is to be carried out is providing a breakdown of the different activities that require different systems or organizations to perform them. SEAD is commonly comprised of 5 different activities [158]. A summary of these activities is provided in Table 9.

Table 9: SEAD Activities.

1.0 Detect	4.0 Target Assignment
2.0 Identify	5.0 Weapon Control
3.0 Correlate/Track	

These higher level activities can then be broken down into lower level functions that must be completed. For example, Weapon Control includes battle damage assessment and possible removal of the target from the target list. An activity flow diagram similar to that found in Ref. [68] is included as Figure 71. The diagram provides a functional breakdown of the higher level SEAD activities. It also includes a mapping of the different systems available for different function tasking. The systems listed in Figure 71 are representative of different system types. While many more systems of a given type can be included in an architecture (*e.g.*, more than one type of fixed-wing fighter such as an F-35, or another airborne surveillance aircraft such as the E-8), for purposes of this study one system from each type is chosen. Table 10 provides an overview of the different system/system types for further clarity. Listed in parentheses next to each available system is the number of functions from Figure 71 the respective system can perform. Table 11 also provides a mapping of the different functions each system can perform.

Table 10: SEAD Systems & Number of Supported Functions.

Available System	System Type
F/A-18 Hornet (5)	Fixed-Wing Fighter/Attacker/Bomber
AH-64 Apache (5)	Attack Helicopter
X-47B (7)	Unmanned Combat Aerial Vehicle
EA-6B Prowler (5)	Electronic Warfare
M252 Mortar Crew (4)	Indirect Fire
DDG (1)	Naval Surface Fire Support
SOF (6)	Special Operations Forces
E-2 Hawkeye (5)	Airborne Surveillance
Intel Satellite (5)	Space-based Sensor
CVN (10)	In-theater Command & Control
Central C2 (10)	Centralized Command & Control

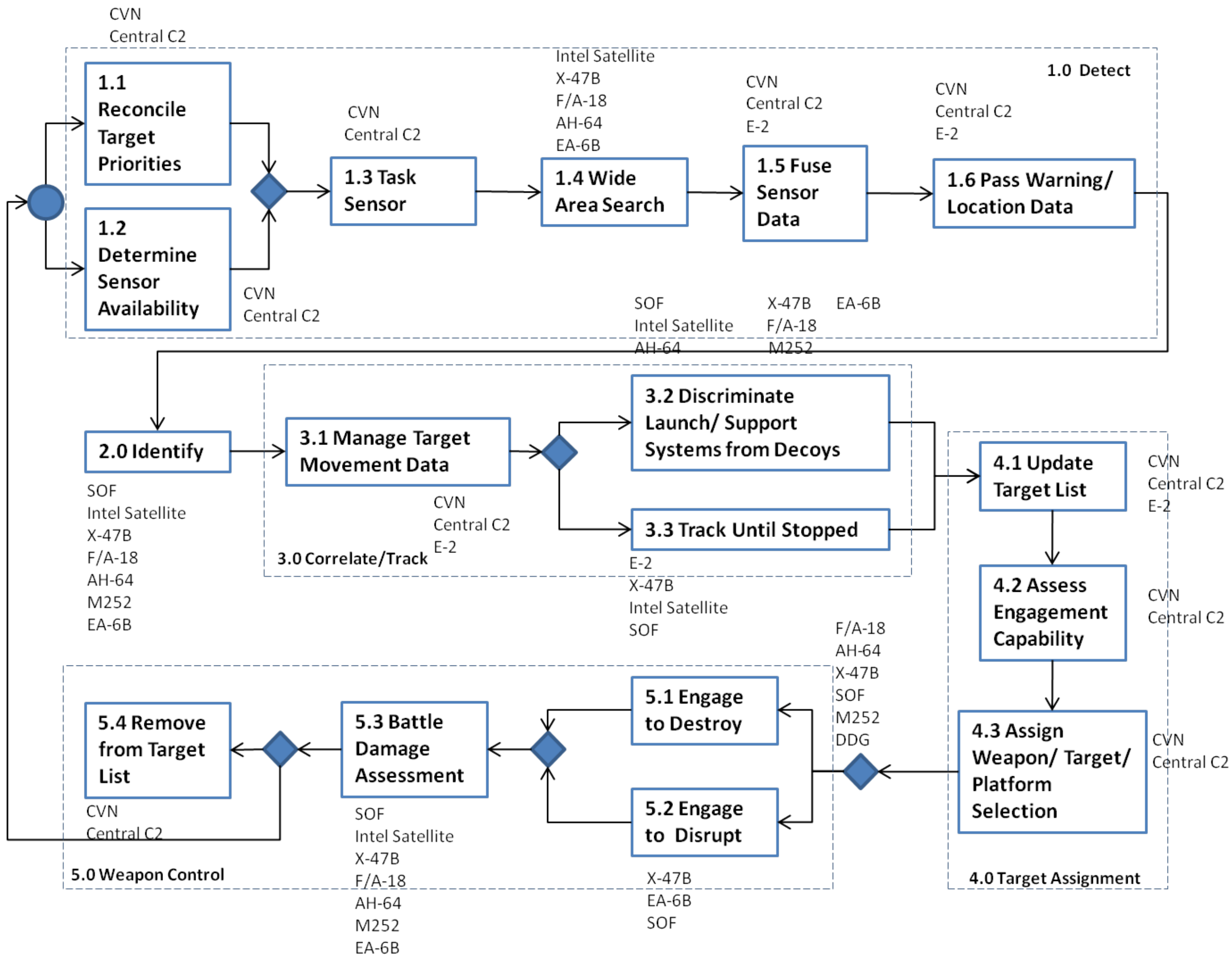


Figure 71: Activity Flow Diagram with System-to-Function Mapping.

Table 11: Available System to Function Mapping.

	1.1	1.2	1.3	1.4	1.5	1.6	2.0	3.1	3.2	3.3	4.1	4.2	4.3	5.1	5.2	5.3	5.4
F/A -18 Hornet				**X**			**X**		**X**					**X**		**X**	
AH-64 Apache				**X**			**X**		**X**					**X**		**X**	
X-47B				X			X		X	X				X	X	X	
EA-6B Prowler				X			X		X						X	X	
M252 Mortar							X		X					X		X	
DDG														X			
SOF							X		X	X				X	X	X	
E-2 Hawkeye					X	X		X		X	X						
Intel Satellite				X			X		X	X						X	
CVN	**X**	**X**	**X**		**X**	**X**		**X**			**X**	**X**	**X**				**X**
Central C2	**X**	**X**	**X**		**X**	**X**		**X**			**X**	**X**	**X**				**X**

Based on the activity activity flow diagram given as Figure 71, a total of 5 distinct needlines exist that comprise the resource state space. They are presented in Table 12, and represent requirements that all alternative architectures must meet. The resource state space is heterogeneous, with $\gamma = 0.67$. For this resource state space, $H(\gamma)$ is 0.92 bits, and this value will be used in the RSC calculation for each alternative, along with the CNE measure of the alternative's resource sharing network.

Table 12: Characterization of SEAD Needlines Using Resource State Specifiers.

Needline Number	Needline Description	Dimensionality	Frequency	Potential
N1	Search & Detect Tasking and Updates	1	1	1
N2	Target ID Data	0	0.5	1
N3	Target Tracking Data & Updates	1	1	1
N4	Target Assignment & Target List Updates	0	0	1
N5	Weapons Payload Status & Engagement Updates	0	0.5	1

In order to calculate RPC later, required IOLs must be specified between the systems that will be included in the analysis. Table 13 provides a summary of this, where dashed numbers represent the possible ranges of IOLs that can exist between a pair of systems.

Table 13: Interoperability Levels for Blue SEAD Force.

System	F/A-18	AH-64	X-47B	EA-6B	M252	DDG	SOF	E-2	Intel Satellite	CVN	Central C2
F/A-18	2	1-2	1-3	2	1	1-2	1-2	2-3	0	5	2
AH-64		2	0-3	1-2	1	0	1-2	1-3	0	1	2
X-47B			2	1-3	0	0	0	2-3	0	5	4
EA-6B				2	2	1-2	1-2	2-3	0	5	2
M252					2	0	2	2	2	1-2	2
DDG						2	1-2	1-2	0	1-2	2
SOF							2	1-2	1-2	1	2
E-2								2	0	5	2
Intel Satellite									0	0	3
CVN										2	2
Central C2											2

The Universal Naval Task List (UNTL) serves as the source for the SEAD MoEs. Using the UNTL, the following top-level SoS requirements are identified [47]:

1. Maximize the % of enemy air defenses and capabilities that are either disrupted or destroyed.

2. Minimize the % of friendly air losses due to enemy air defenses.

7.2 Step 2: Define Alternative System Portfolios

Since many different systems are capable of carrying out the required tasking presented in Figure 71, one of the key questions that arises in architectural design is a combinatorial one. The architect must decide which is the best portfolio of systems to carry out the required tasking. Using Figure 71, three alternative system portfolios are constructed from the list of available systems. They are depicted in Table 14.

Table 14: Alternative SEAD System Portfolios.

Alt.1	Alt.2	Alt.3
Central C2	CVN	CVN
SOF Team	F/A-18	F/A-18
Intel Satellite	X-47B	E-2
-	-	EA-6B
-	-	DDG
-	-	AH-64
-	-	M252

The system portfolios are chosen so that all 11 system types are represented, though many more combinations of system portfolios can be evaluated. When defining system portfolios it is important to ensure feasibility, meaning that all of the required functionality is covered by at least one combination of the systems included in the portfolio.

7.3 Step 3: Generate Feasible Architecture Alternatives

The aim of NCW is the generation of additional combat power from platforms that are linked as part of a shared awareness network [12]. This is in contrast to traditional, platform-centric warfare. As Perry explains, in platform-centric warfare “One must mass force to mass combat effectiveness because each weapon system acts independently, whereas in network-centric warfare effects are massed, rather than force"[137]. While there are assumed benefits to NCW, there are possible downsides as well. As we have seen, there is increased complexity due to the networking of individual platforms. This complexity may negatively impact performance. For example, with many systems sharing the same communications network, there exists a greater possibility for information overload [137]. Perry goes on to state:

> Traditional measures of effectiveness (MOEs) usually ignore the effects of information and decision making on combat outcomes. In the past, C4ISR operations have been analyzed separately using measures of performance (MOPs). The effects of changes in C4ISR operations on combat outcomes have been inferred rather than directly assessed, and therefore the quantifiable link between variations in C4ISR capabilities and combat outcomes has been relatively difficult to assess.

From this, it becomes clear that to fully and accurately assess the effectiveness of a network-centric SoS architecture means that the M&S environment must be able to capture any potential differences in effectiveness between platform-centric and network-centric architectures [108, 124]. Before this can be accomplished, however, we must first understand the different types of SoS collaborations that distinguish platform-centric architectures from network-centric ones. Collaboration can be defined as "a process in which individuals work together to achieve a common goal. Shared information is an essential ingredient to ensure effective collaboration" [137, 138]. In general, more effective collaboration and information sharing has a direct impact on the speed and quality of decision making that occurs.

7.3.1 SoS Collaboration Categories

The task hierarchy and system-to-function mapping shown in Figure 71 is the best place to begin when attempting to understand the different patterns of collaboration that define a SoS architecture. Using Figure 71, it is clear that three distinct types of collaboration can take place between systems that are part of a SoS. These categories are described as the following:

- Category 1 — Required Task-to-Task Collaboration: Resource exchanges occur between systems performing sequential tasking.
- Category 2 — Shared Task Collaboration: Resource exchanges occur between systems with shared tasking.
- Category 3 — Non-specific Collaboration: Resource exchanges occur between systems performing indirectly linked, non-sequential tasking.

An example using the SEAD task hierarchy and system-to-function mapping is provided as Figure 72. Figure 73 gives a visual description of a platform-centric SoS architecture versus a network-centric SoS architecture. In the platform-centric case, collaborations are usually limited to Category 1 collaborations and C2 is hierarchical and centralized. For example, in Figure 73, all communications and coordination are routed through the central C2 node, which is the aircraft carrier (CVN). In contrast, the distributed, fully networked SoS architecture is decentralized in structure. In addition to Category 1 collaborations, Category 2 collaborations are usually present. Category 3 collaborations may occur as well. While the CVN retains the primary C2 functions within the architecture, more direct communications and coordination can be achieved between other systems within the SoS. There are several possible technical implementations for this type of architecture.

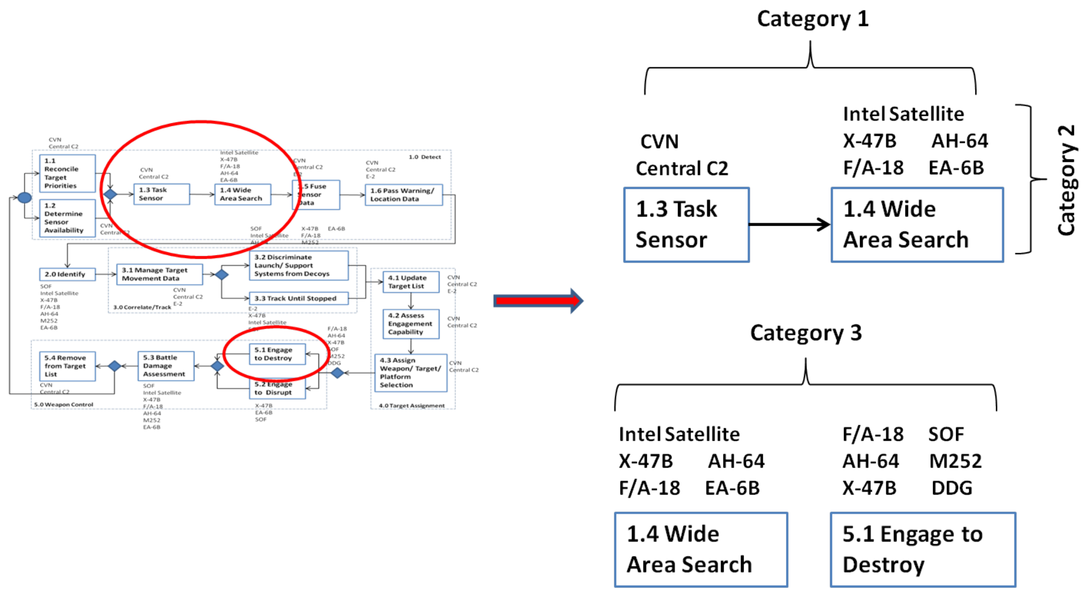


Figure 72: SEAD Example of SoS Collaboration Categories.

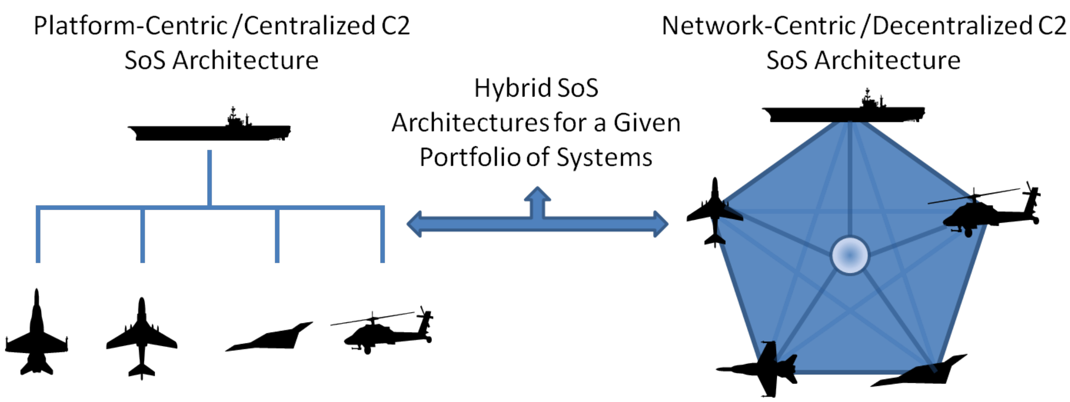


Figure 73: Platform-Centric SoS Architecture vs. Network-Centric SoS Architecture.

7.3.2 Design Space of Alternative SoS Architectures

Keeping in mind that an architecture is defined as the structure of components, their relationships, and the principles and guidelines governing their design evolution over time [64], and that each alternative architecture can be uniquely represented by DoDAF or elements thereof, it follows that alternative military SoS architectures can also be generated by manipulating baseline DoDAF products. For a given capability need, SoS architectures can differ in the following ways:

- *Operational/Process Variations:* Changes made to the choice of tasks/activities used to provide capabilities and changes in the sequencing and timing. Figure 74 gives a generic example. In Figure 74, there is a difference in when tasks in the second sequence can be bypassed and there is also a difference in the sequential/parallel tasks that comprise the second sequence itself. The DoDAF Event-Trace Description, referred to as the OV-6c, is used to trace the actions in a scenario or sequence of events. The CV-6, the Capability to Operational Activities Mapping, portrays the mapping between the required capabilities and the operational activities that those capabilities support.

- *System & Technology Variations:* Composed of changes to the specific tasks or activities that specific systems within the architecture can support. Total distribution of functionality across the SoS designs multiple single points of failure, therefore there is often some form of redundancy or duplication of functionality. When this occurs, this opens up the space for possible system-to-system interactions. Force structure variations are also possible, meaning possible changes to the specific number and types of constituent systems included in the SoS. Included systems can range from older legacy systems to newer systems specifically adapted for operating as part of the SoS. Also of note, a top-down CBA may identify novel applications for existing systems in order to achieve desired operational effects. An example is the use of a CVN as a rapidly deployable emergency power station during disaster aid and relief efforts. Individual systems may also vary in the specific technologies and technical standards they employ, which affects their cost, performance, and possibly their availability. For example, the Army's Apache Block III program is a planned upgrade to AH-64D Longbow Apache helicopters expected to improve performance, situational awareness, lethality, survivability, interoperability, and the prevention of friendly fire incidents. This requires each Apache to go the factory for hardware changes while software changes are installed in the field [87]. Much of this information is contained within the various models that comprise the Systems Viewpoint in DoDAF V2.0.

- *Interface & Network Variations:* Changes to the way in which resources such as data and information is exchanged between systems. This includes changes to the specified IOL, changes in collaboration ranging from platform-centric to network-centric patterns of collaboration, and changes to the technical implementation of the interfaces used to realize the distribution and sharing of resources. Much of this information is contained with the various models that comprise the Services Viewpoint in DoDAF V2.0.

- *Organizational Variations:* Changes to the organizations responsible for particular systems, processes, and activities. Changes to the structure of the relationships among organizations can occur as well. OV-4, the Organizational Relationships Chart, provides the organizational context, role, or other relationships among organizations.

Table 14 specifies the three different system groupings contained in each candidate system portfolio. It is expected that the performance of each system portfolio will be affected differently as the architecture that specifies the relationships between systems

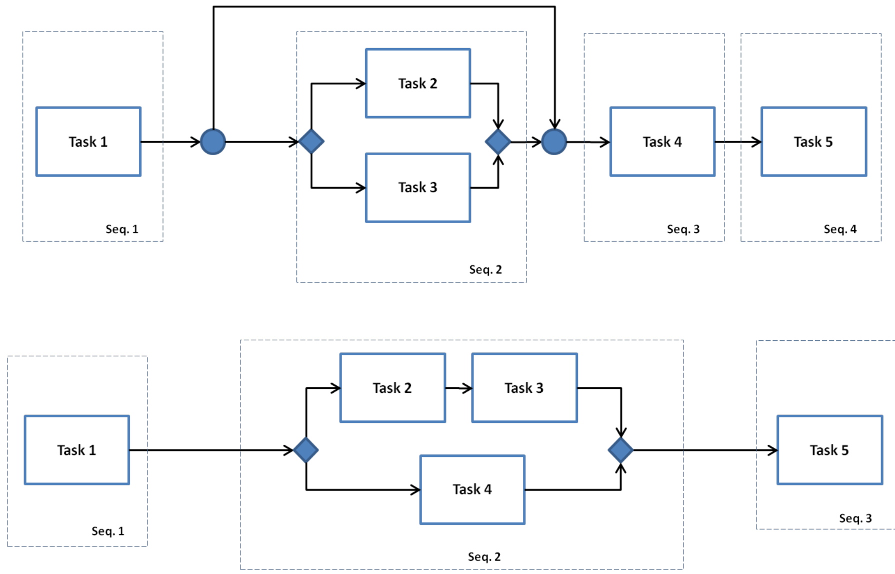


Figure 74: Example Process Sequence Changes.

within a particular portfolio changes. For a given portfolio of systems, alternatives are generated by allowing variations in the patterns of collaboration between systems as well as the level of interoperability, or IOL. This can be seen in Figure 75, where the system portfolio is represented in a vector format, with each number representing the presence (or absence in the case of a zero) of the numbers of a particular system type. The matrices in the Collaboration & Interoperability Variations column are the Resource Processing Matrices, with each entry in the matrix representing the specified IOL. Alternatives are also generated by allowing the force structure of Blue SEAD assets to vary. Figure 76 shows the effect of variations in force structures. In order to limit the amount and complexity of the M&S environments that need to be created, neither operational/process variations nor technology variations are included in this analysis.

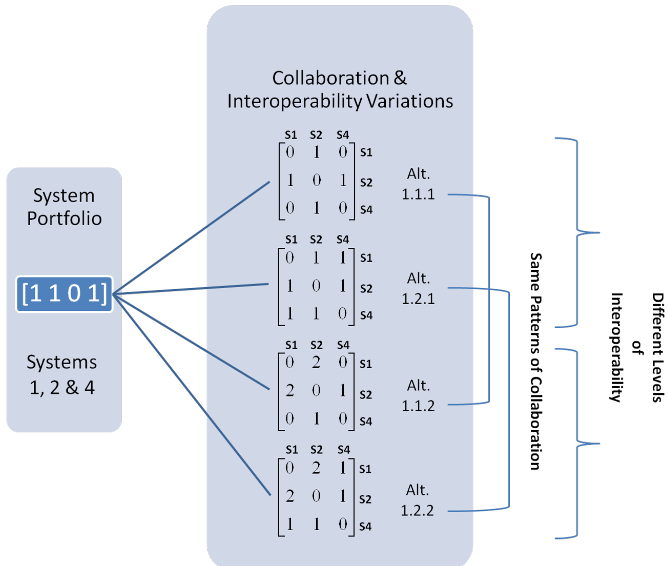


Figure 75: Defining Alternatives: Variations in Collaboration Patterns & Interoperability Levels.

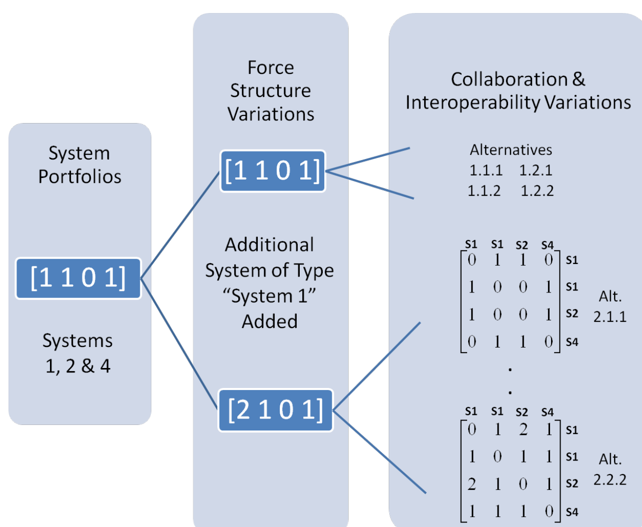


Figure 76: Defining Alternatives: Variations in Force Structures.

7.4 *Step 4: Develop M&S Environments and Perform Cost Analyses*

7.4.1 SEAD Mission Scenario Development

Intelligence reports confirm that country Red has successfully developed a nuclear theater ballistic missile (TBM-N) capability with support from neighboring country Orange. This is in clear violation of international peace treaties. International sanctions have failed to deter Red's military and political leaders from bringing the TBM-N capability to IOC. This has lead to the immediate destabilization of the entire region and has sever international implications.

Country Blue has tasked its military commanders with eliminating Red's TBM-N capability by destroying the co-located TBM launch and nuclear fuel processing facilities. These facilities have been fortified and hardened, requiring the use of heavy bombing payloads from Long Range Strike (LRS) assets. Red possesses an IADS consisting of the following:

- Early Warning Radars (EWRs)
- Surface-to-air Missiles (SAMs)
- Anti-Aircraft Artillery (AAA)
- Centralized Command & Control (C2) Center

Figure 77 is a visual depiction of Red's EWR, SAM, and AAA assets. The SEAD AOR/JOA is shown in Figure 78. The inland location of the target facilities prohibit direct naval cruise missile strikes on these facilities, but naval surface fire support can be brought to bear on some IADS assets. The time critical nature of the mission coupled with the sophistication and lethality of Red's IADS requires the SEAD within the AOR/JOA shown in Figure 78. Pre-planned SEAD in the AOR/JOA will be conducted in two phases. The goal of Phase I is locate, destroy, and disrupt Red mobile EWRs. Intel suggests that Red's inventory has been supplemented by an unknown number of Orange-supplied EWRs. The EWRs employ camouflage and concealment to help evade detection, especially by satellite imagery. The mobile EWRs also frequently shift locations to ensure their survivability, since they have very low defensive capabilities. After the locations of EWRs within the AOR/JOA have been ascertained and as many EWRs neutralized, Phase II will commence. Phase II consists of Blue forces entering the Red IADS engagement zone with the goal of neutralizing Red SAM sites. Successful completion of Phases I & II will allow successful LRS operations.

Figure 77: Systems in the Red IADS.

7.4.2 Phase I M&S: Modeling the Effects of Collaboration

7.4.2.1 ARCNET Development

Now that alternative SEAD SoS architectures can be defined, the next task is to determine a way to estimate the benefits of increased knowledge resulting from the

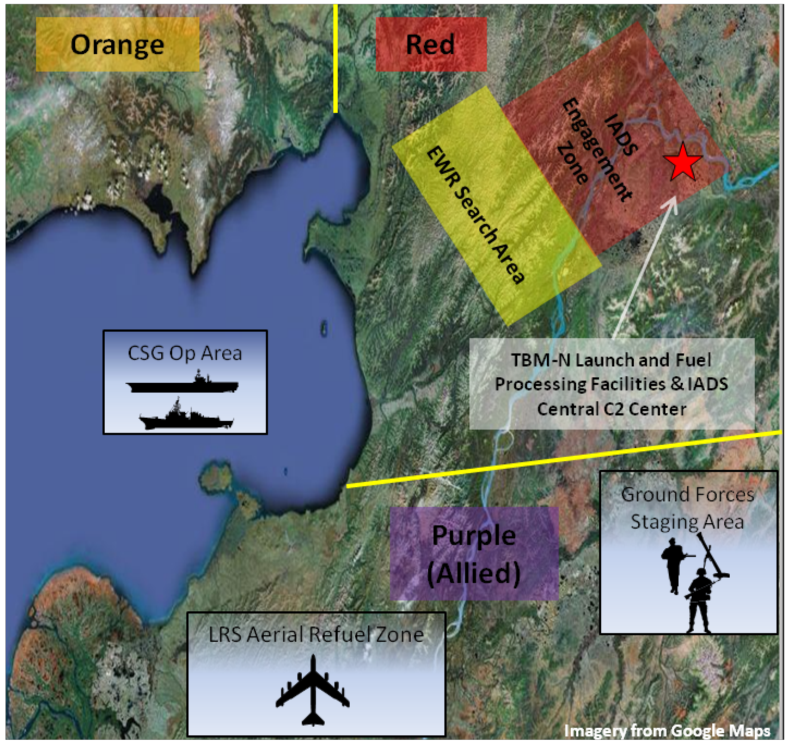


Figure 78: SEAD Mission Scenario AOR/JOA.

changes in collaboration patterns, IOL, and force structure. To complete this task, a method proposed by Perry [137] is adapted for use. Perry's method relies upon Information theory, with information entropy concepts developed by Shannon to assess the "amount" of knowledge available in a C2 system [137, 147].

The first step in Perry's method is to determine the primary sources of uncertainty that affect mission requirements and MoEs, then to quantify this mission uncertainty using a probability distribution. Once the distribution $f(x)$ is assumed, Information entropy provides a measure of the average amount of information in the probability distribution. Information entropy (also commonly referred to as *Shannon entropy*) is based on the notion that the amount of information in the occurrence of an event is inversely proportional to the likelihood that the event will occur. The Information entropy of $f(x)$ can be assessed by applying Equation 42, which is the differential form of Shannon's Information entropy equation.

$$H(x) = -\int_{\infty}^{\infty} ln\left[f(x)\right] f(x)dx \tag{42}$$

For the SEAD AOR/JOA mission scenario, the primary source of uncertainty that must be modeled occurs in Phase I of the mission. This is where mission planners face uncertainty surrounding the number and location of EWRs within the AOR/JOA. This uncertainty must be resolved since the EWRs will pass radar cueing data to the SAM sites, allowing them first shot opportunity. This directly impacts the probability of successfully engaging Red SAM sites and the success of the subsequent LRS mission. It is assumed that collaboration among systems in the SEAD SoS could help reduce this uncertainty, perhaps through coordinated search efforts and the sharing of information to perform data fusion.

To determine $f(x)$, the EWR search area is divided into individual sensor grids. The assumption is made that EWRs must maintain a minimum separation distance from each other to maintain effective search coverage. It is also assumed that the individual sensor grids are small enough such that multiple EWRs will not be co-located within a single search grid. As a consequence of these assumptions, locating an EWR in one grid is independent of finding EWRs in other grids. This allows modeling the search as a Poisson process, where the number of events occurring in a time interval of length t has a Poisson distribution with mean λt. As a result an exponential distribution is chosen for $f(x)$. Equations (43-46) show the probability density function, cumulative distribution function, expected value, and variance, respectively.

$$f(x) = \lambda e^{-\lambda x} \tag{43}$$

$$F(x) = 1 - e^{-\lambda x} \tag{44}$$

$$E(X) = \frac{1}{\lambda} \tag{45}$$

$$Var(X) = \frac{1}{\lambda^2} \tag{46}$$

$E(X)$ can be interpreted as the average number of sensor grids that must be searched between detections. As the number of EWRs increase, then E(X) should decrease and vice versa. Next, an independent sensor coverage parameter, λ_i is defined for each Phase I search platform. Table 15 illustrates the different factors that affect this parameter. Also, Equation (47) shows how these factors can be combined to calculate λ_i for each search aircraft. Equation (48) shows that the combined sensor coverage parameter is simply the summation of the independent sensor coverage parameters for each asset taking part in the Phase I search for EWRs.

Table 15: Aircraft & Satellite Independent Sensor Coverage Factors.

Search Aircraft (X-47B, F/A-18, AH-64)	Intel Satellite
Search Area (A)	Resolution
Sweep Width (s)	Availability
Search Speed (v)	Imagery Processing Time
Effective Search Time (T)	Occlusion From Weather or Other Obstructions

$$\lambda_i = 1 - e^{-(svT/A)} \tag{47}$$

$$\lambda = \sum \lambda_i \tag{48}$$

It is important to note that the exponential distribution also possesses the *memoryless property*, meaning that if the random variable X measures the time until a certain event occurs and the event has not occurred by time x_o, the *additional* waiting time for the event to occur beyond x_o has the same exponential distribution as X. If any of the original assumptions are changed, the memoryless property may not hold, and a more suitable distribution should be chosen [91].

Once $f(x)$ has been defined, the next step is to create a mapping of entropy onto a [0,1] knowledge scale by selecting an upper bound on the entropy. This is done by first applying Equation (42) to the recently defined exponential distribution, $f(x)$. This results in Equation (49).

$$H(x) = ln\left(\frac{e}{\lambda}\right) \tag{49}$$

This suggests the following definition for the knowledge associated with the exponential distribution:

$$K(x) = \begin{cases} 0 & \text{if } \lambda < \lambda_{min} \\ ln\left(\lambda/\lambda_{min}\right) & \text{if } \lambda_{min} \leq \lambda < e\lambda_{min} \\ 1 & \text{if } \lambda \geq e\lambda_{min} \end{cases} \tag{50}$$

Where:

$$e \times \lambda_{min} = 1 \tag{51}$$

And:

$$\lambda_{min} = \frac{1}{e} = 0.368 \tag{52}$$

A plot of the normalized knowledge function is presented as Figure 79. Figure 79 reflects the relative amount of information about the number and location of EWRs that can be gained from independent search operations during Phase I. Obviously, as more assets are engaged in the search, combined search coverage increases along with knowledge.

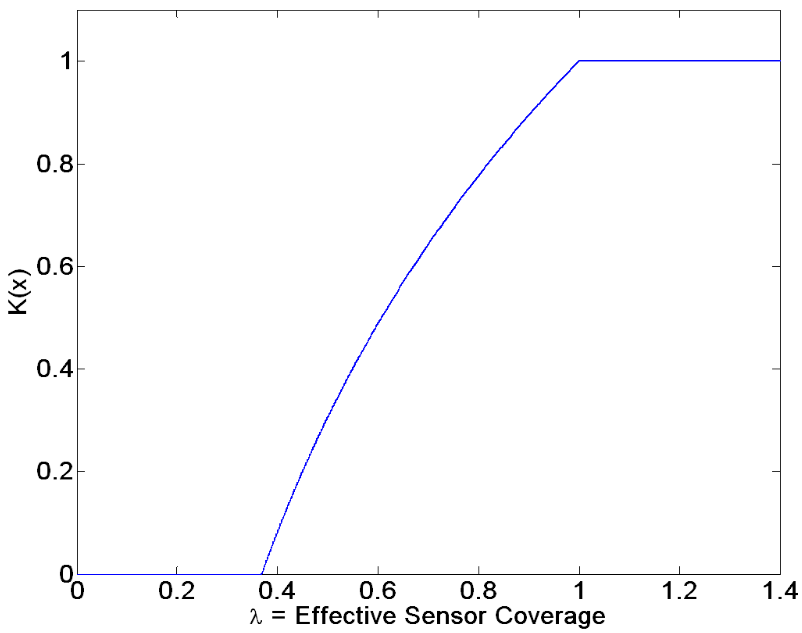


Figure 79: SEAD Phase I Normalized Knowledge Function.

Once the amount of knowledge that can be gained from independent operations can be estimated, the next step is to model the impact of collaboration. Perry's method assumes a statistical reliability model to be an appropriate model. The collaboration between a pair of systems takes the general form of Equation (53), where $r(s)$ is called the failure rate function and is dependent on the nature of the collaboration.

$$c_{ij}(t) = 1 - e^{-\int_0^t r(s)} ds \tag{53}$$

When there is no time to collaborate, *i.e.*, $t = 0$, then $c_{ij}(t) = 0$. Additionally, the time at which successful collaboration occurs between two systems depends on the form of the failure rate function for that collaboration. A constant is selected so that earlier successful collaboration can be modeled by simply increasing the constant value. This leads to the following form of the previous equation:

$$c_{ij}(t) = 1 - e^{-\theta t} \text{ for } t \geq 0 \tag{54}$$

Referring back to Table 7, the IOL that exists between the two systems serves as a measure of the amount of information or services that can be exchanged directly and satisfactorily between them. Table 16 shows the mapping of IOL to θ used for Phase I of the SEAD mission. Instead of simply assigning a single value for θ at each IOL, the value of θ is given a normal distribution and standard deviation. Figure 80 is a plot of the collaboration reliability curves, using the mean values of θ for ease of presentation. The mapping was chosen so that there would be diminishing returns as IOL levels increase. This effect can be seen in a preliminary SEAD M&S study conducted in Ref. [17]. More research is needed in this area to more accurately determine how changing IOL affects collaboration reliability, and how much variability there is at each IOL.

Table 16: IOL to Reliability Constant Mappings.

IOL	θ (mean)	Std. Dev.
0	0	0
1	0.40	0.10
2	0.60	0.10
3	0.75	0.05
4	0.90	0.05
5	0.98	0.01

Since Equation (54) is a cumulative probability, the probability density function can be calculated:

$$f_{ij}(t) = \theta e^{-\theta t} \tag{55}$$

The probability density function is an exponential distribution with $1/\theta$ being the mean time for systems i and j to collaborate with a variance of $(1/\theta)^2$. The entropy calculation for the exponential distribution with parameter θ is:

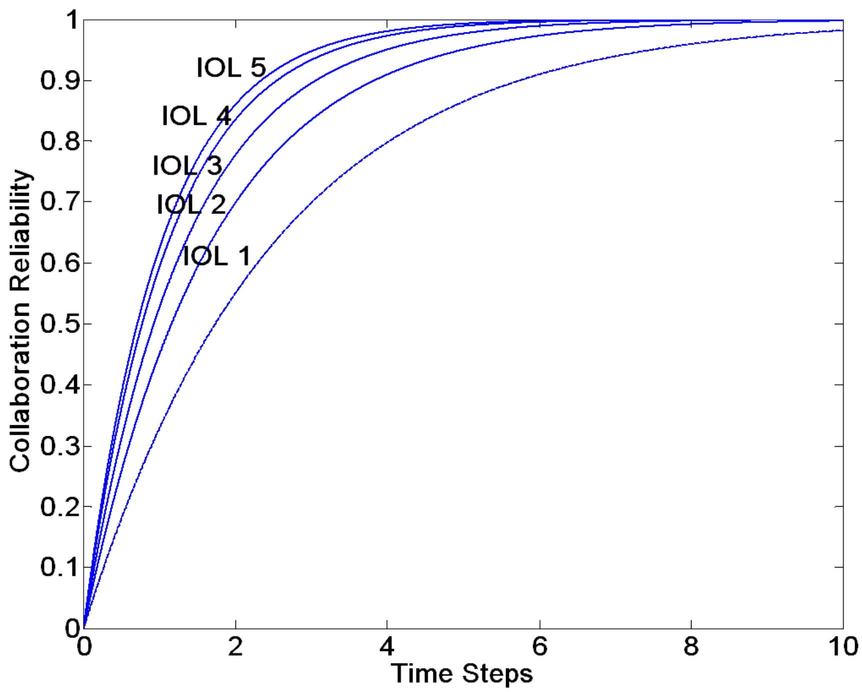


Figure 80: SEAD Phase I Collaboration Reliability Curves.

$$H(t) = -\int_{t=0}^{\infty} ln\left[\theta e^{-\theta t}\right]\theta e^{-\theta t}dt = 1 + ln\left(\frac{1}{\theta}\right) = ln\left(\frac{e}{\theta}\right) \tag{56}$$

Careful study of this formulation yields the following observations:

- The entropy varies with the variance of the distribution as should be expected.
- As $(1/\theta)$ increases (θ decreases), $H(t) = ln(e/\theta)$ also increases.
- Entropy is unbounded for the distribution, which is true for all continuous distributions.

The collaboration entropy function can now be used to develop a measure of knowledge by assessing the "certainty" in the density function. An approximate upper bound is assigned to H(t), the equivalent to assigning a maximum expected time to complete a collaboration. Letting $(1/\theta)_{max} = \theta_{min}$ represent the maximum expected time, then a measure of certainty or knowledge can be written as:

$$K(t) = ln\left(\frac{e}{\theta_{min}}\right) - ln\left(\frac{e}{\theta}\right) = ln\left(\frac{\theta}{\theta_{min}}\right) \tag{57}$$

Perry notes that $K(t)$ is a dimensionless quantity and therefore can be used directly to influence combat MoEs. He also states that it is desirable, however, for the measure of knowledge to be normalized. This can be accomplished by noting that when $\theta = \theta_{min}$, $K(t) = ln(1) = 0$ and when $\theta/\theta_{min} = e$, $K(t) = ln(e) = 1$. This suggests the following definition for the knowledge gained from the collaboration between systems i and j:

$$K_{ij}(t) = \begin{cases} 0 & \text{if } \theta < \theta_{min} \\ ln\left(\theta/\theta_{min}\right) & \text{if } \theta_{min} \leq \theta < e\theta_{min} \\ 1 & \text{if } \theta \geq e\theta_{min} \end{cases} \tag{58}$$

Where the following values are chosen for the SEAD scenario:

$$\theta_{max} = e \times \theta_{min} = 0.95 \tag{59}$$

And:

$$\theta_{min} = \frac{\theta_{max}}{e} = \frac{0.95}{e} = 0.35 \tag{60}$$

In Equation (58), for small values of θ, the mean and variance are large, thus implying great uncertainty and therefore little knowledge. For large values of θ, the opposite is true and therefore considerable knowledge is gained. In this way, $K_{ij}(t)$ models the positive effects of having more time, on average, to collaborate. A plot of the collaboration knowledge curve is provided in Figure 81.

After this, the next step in Perry's method is to determine a total system collaboration factor that accounts for all pairs of collaborating systems. In his example,

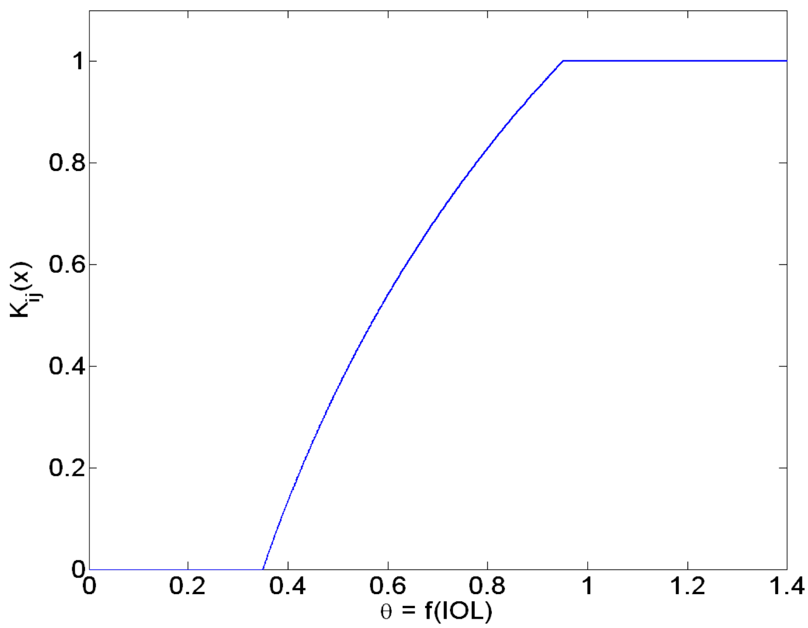


Figure 81: SEAD Phase I Collaboration Knowledge Curve.

Perry uses an inverse reliability model for sequential overall system collaboration. This results in the following:

$$K_M(t) = 1 - \prod_{[i,j]} K_{ij}(t) \tag{61}$$

Once $K_M(t)$ is calculated, the effects of collaboration are represented using the following linear model:

$$K_C(\lambda) = K_M(t)\,[1 - K(\lambda)] + K(\lambda) \tag{62}$$

Equation (61) assumes the that the collaboration effect from each collaborating pair is equal in value. This method is not directly applicable to the SEAD mission for the following reason. As IOL increases, so does the value of θ. Larger values of θ equate to larger $K_{ij}(t)$ between systems. For a given number of systems, using Equation (61) would result in a smaller value of $K_M(t)$ and thus a smaller value of $K_C(\lambda)$. For example, when examining a SoS with two collaborating pairs, the SoS with lower quality collaboration would rate higher:

$$K_M(t) = 1 - \prod_{[i,j]} K_{ij}(t) = 1 - (0.8)(0.8) = 0.36 \tag{63}$$

$$K_M(t) = 1 - \prod_{[i,j]} K_{ij}(t) = 1 - (0.5)(0.5) = 0.75 \tag{64}$$

For this analysis, a modified method of combining the effects of localized collaboration is used instead. The modified method begins with defining an $n \times n$ collaboration matrix, CK. Each entry of CK, or ck_{ij} is equal to the $K_{ij}(t)$ between the pairs of systems that collaborate to detect, identify, and track EWRs during Phase I of the SEAD mission.

$$CK = \begin{bmatrix} 0 & ck_{1,2} & \dots & & ck_{1,n} \\ ck_{1,2} & 0 & \dots & & \vdots \\ \vdots & & \ddots & & \vdots \\ ck_{1,n} & & \dots & & 0 \end{bmatrix} \tag{65}$$

Next, the maximum absolute eigenvalue of CK ($\lambda_{max}^{(CK)}$) and the number of systems (n) is used to determine $K_M(t)$:

$$0 \leq K_M(t) = \frac{\lambda_{max}^{(CK)}}{(n-1)} \leq 1 \tag{66}$$

This is analogous to determining the CNE, where the PFE is normalized by the number of nodes. Since the CK matrix consists of actual systems and not *system types*, the diagonals of the CK matrix are zero (collaboration is defined as occurring with another external system). Normalization is therefore performed by dividing by $(n-1)$ nodes, instead of n.

The last step is to include the negative effects of collaboration in the model. Perry attempts to capture these negative effects as the complexity that results as the total number of connections between systems increase. However, Perry acknowledges that "complexity alone, as defined by the number of connections in a network, is clearly not enough to assess the effectiveness of network-centric operations" [137]. Therefore, instead of using the number of connections, the resource processing matrix RP that is used to determine RPC is used instead. Similar to what was done for the collaboration matrix CK, a normalized network complexity factor C_n can be calculated in the following manner:

$$C_n = \frac{(1/IOL_{max}) \times \lambda_{max}^{(RP)}}{(n-1)} \times \left(1 - \frac{1}{\sqrt{n-1}}\right) \tag{67}$$

$$0 \leq C_n \leq 1 \tag{68}$$

Estimating the effects of collaboration complexity in this way is in keeping with Perry's method, since RPC takes into account interoperability, which strongly influences network complexity. The maximum eigenvalue is normalized by the $(n-1)$ numbers of systems. By taking the normalized eigenvalue of the RP matrix, a weighted connection density is obtained. This weighted connection density can then be corrected to avoid penalizing architectures with small force structures the same as architectures with large ones. For example, without the correction factor, a network consisting of two systems with maximum interoperability would have the same normalized network complexity factor as a network consisting of 100 nodes all operating at max IOLs. Once C_n is calculated, a logistic curve is used to simulate a nonlinear trend and obtain a value of $g(C_n)$. Once the normalized complexity of the network increases past a certain threshold value, the negative effects of collaboration accumulate much more rapidly.

$$g(C_n) = \frac{e^{-7.5+9.5C_n}}{1+e^{-7.5+9.5C_n}} \tag{69}$$

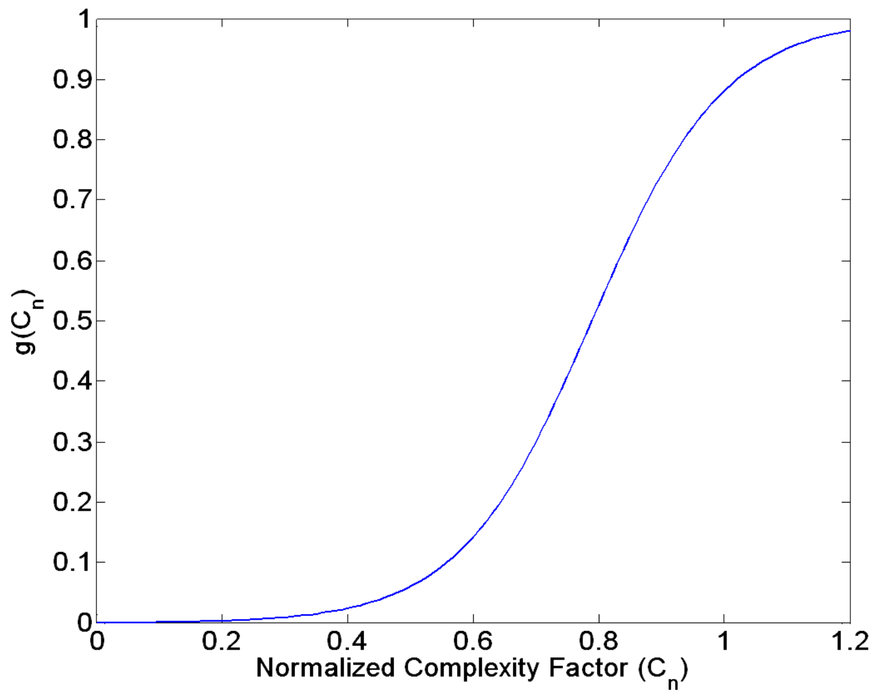


Figure 82: Complexity Logistic Curve Mapping.

The final equation, taking into account complexity effects is then:

$$K_{CC}(\lambda) = [1 - g(C_n)] \times \{K_M(t)[1 - K(\lambda)] + K(\lambda)\} \tag{70}$$

Equation (70) can be used to determine the total knowledge obtained by the systems employed during Phase I of the SEAD mission. K_{CC} is then directly used as the probability of locating an EWR in the search area. Once located, the EWR is engaged by Blue assets.

Overall, incorporation of Perry's method into the M&S environment meets the objective of adequately assessing the impact of net-centricity on combat outcomes. Implementation of Perry's method results in the creation of an Architecture Resource-based Collaborative Network Evalulation Tool (ARCNET) that can be reused for many different types of mission scenarios.

7.4.3 Phase II M&S: Simplified Engagement Model Development

Now that the mission scenario has been fully described, the next step is to begin the task of M&S in order to determine: 1) the % of IADS assets suppressed and 2) the number and % of Blue units lost during the engagement [47]. These are the MoEs identified from the UNTL for assessing the success of a SEAD operation. M&S of the mission scenario is based on the following assumptions developed by Barkdoll *et al.* for modeling SEAD [25]:

1. Pre-planned SEAD is being employed by Blue in the AOR/JOA against a fully operational Red IADS.
2. Red is defending their launch and nuclear processing facilities with several SAMs. These SAMs have overlapping engagement envelopes that create a single engagement zone.
3. Blue attackers/weapons must enter the engagement zone to reach their targets. The time spent inside the engagement zone is Blue's vulnerability window.
4. Red engagement radars are co-located with Red SAM batteries.
5. Blue forces within the engagement zone can be automatically detected by SAM engagement radars, but each SAM can only execute a certain number of engagements at a time, as determined by the SAM engagement radar saturation rate.
6. Saturation of EWRs and ground unit detectors can occur when the number of Blue forces exceed the EWR saturation rate.
7. Red does not present a significant fighter threat to Blue.
8. Blue attackers start beyond the radar horizon (undetected).

9. Blue LRS bomber aircraft are separated from the Blue attackers and remain outside the Red engagement envelope during Phase I. LRS assets are scheduled to enter Red airspace upon commencement of Phase II.

10. Both Red & Blue forces have sufficient weapons/munitions to cover attacks and re-attacks for the entire duration of the vulnerability window.

Using these assumptions, a simplified engagement model (SEM) is created. Inputs to the SEM include the relevant size and disposition of both Red and Blue forces. The size of each force is determined by the force structure, which details the specific number of each unit taking part in the engagement. Relevant MoPs for both Red and Blue forces that serve as inputs to the SEM include the probability of hitting an enemy with an attack to either destroy or disrupt (P_{Hit}), the probability of surviving an attack if hit ($P_{Survive|Hit}$), and the maximum number of enemy units that can be engaged within a specific time window. Additional M&S inputs to describe enemy radars include the probability of detecting enemy units (P_{Detect}) and a stealth probability ($P_{Stealth}$). $P_{Stealth}$ describes the probability of evading detection, and is also include for Blue SEAD systems as appropriate. Lastly, IOLs may vary between Blue systems also. These M&S inputs are summarized in Tables 17– 19.

Table 17: M&S Inputs for Red IADS.

System	Min # Available	Max # Available	$P_{Survive\|Hit}$	P_{Detect}	P_{Hit}	Saturation Rate
EWR	3	6	0.1	0.75	-	6
SAM Site	2	4	0.1	-	0.5	3

Table 18: M&S Inputs for Blue SEAD Force (Portfolios 1 & 2).

System	Min # Available	Max # Available	Sensor Coverage (λ_i)	$P_{Stealth}$	P_{Hit}	$P_{Survive\|Hit}$	Max Attack Rate Against EWRs
F/A-18	2	6	0.10	0.40	0.60	0.30	3
X-47B	2	6	0.10	0.50	0.60	0.30	3
SOF Team	5	10	-	0.70	0.65	0.40	1
Intel Satellite	1	1	0.30	1	-	1	-
CVN	1	1	-	1	-	1	-
Central C2	1	1	-	1	-	1	-

Table 19: M&S Inputs for Blue SEAD Force (Portfolio 3).

System	Min # Available	Max # Available	Sensor Coverage (λ_i)	$P_{Stealth}$	P_{Hit}	$P_{Survive\|Hit}$	Max Attack Rate Against EWRs
F/A-18	2	4	0.10	0.40	0.60	0.30	3
AH-64	2	4	0.05	0.65	0.50	0.15	2
EA-6B	2	4	0.08	0.40	0.60	0.25	3
M252	3	3	-	0.60	0.70	0.50	1
DDG	1	2	-	0.50	0.30	1	5
E-2	1	1	-	0.40	0.60	0.15	-
CVN	1	1	-	1	-	1	-

Instead of a full factorial Design of Experiments (DOE) for the Red IADS force structure, the fractional factorial DOE in Table 20 is used. The primary reason for this is that each alternative is re-run a certain number of times dependent upon a parameter referred to as the *block size*. The block size is defined as the product of the number of repeat engagements and the size of the Red Force Structure DOE. Consequently, the number of model executions increases dramatically as the numbers of alternatives grow in size with changes in force structure, collaboration patterns, and variations in IOLs. Later, as changes in process sequences and technology are included, even more emphasis must be placed on intelligently sampling and exploring the design space using DOEs and advanced M&S techniques.

Table 20: Red Force Structure DOE.

Red IADS Force Structure	
Number of Early Warning Radars	Number of SAM Sites
6	2
3	4
5	3
5	2
3	2
3	3
6	3
5	4
6	4

7.4.4 M&S Results

Execution of the engagements is stochastic, and random uniform distributions are used for random variable sampling. For example, to determine if an attack is successful, a random number is drawn from a uniform distribution with interval [0,1]. If the selected number is less than the attack probability, then the attack is considered successful. If desired, other distributions may be used in place of the uniform distribution. Each engagement is re-run numerous times over varying operational parameters in order to obtain mean and standard deviations for the engagement MoEs. Table 21 provides a summary of the number of run executions, which vary by portfolio primarily due to the number of force structure variations that are analyzed. The performance of each alternative generated from the three separate system portfolio groupings can be seen in Figures 83–85. (To prevent confusion, it should be noted that the y-axis on Figures 83–85 should be interpreted as 0.4 = 40%, and not 0.4%, for example.) An example force structure legend interpretation is also given in Table 22.

Table 21: SEM Run Execution Summary.

Vulnerability Window = 2			
Portfolio Number	Number of Repeat Engagements	Block Size	Total Number of Run Executions
1	100	900	10,800
2	100	900	32,400
3	100	900	115,200

Table 22: Example Force Structure Legend Interpretation.

System Type	F/A-18	AH-64	X-47B	EA-6B	M252	DDG	SOF Team	E-2	Intel Satellite	CVN	Central C2
# Included	0	0	0	0	0	0	10	0	1	0	1

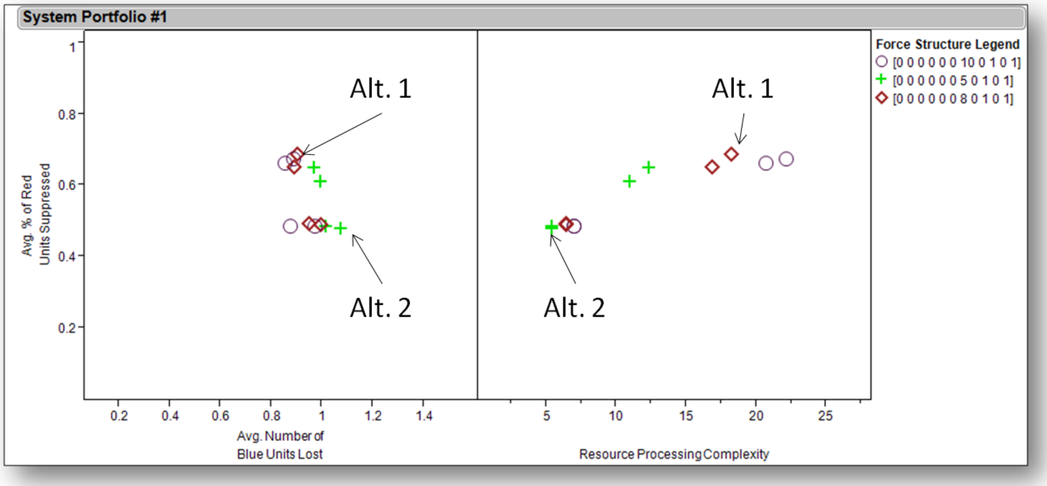


Figure 83: Mission Performance for Alternatives Generated from the 1st SEAD System Portfolio.

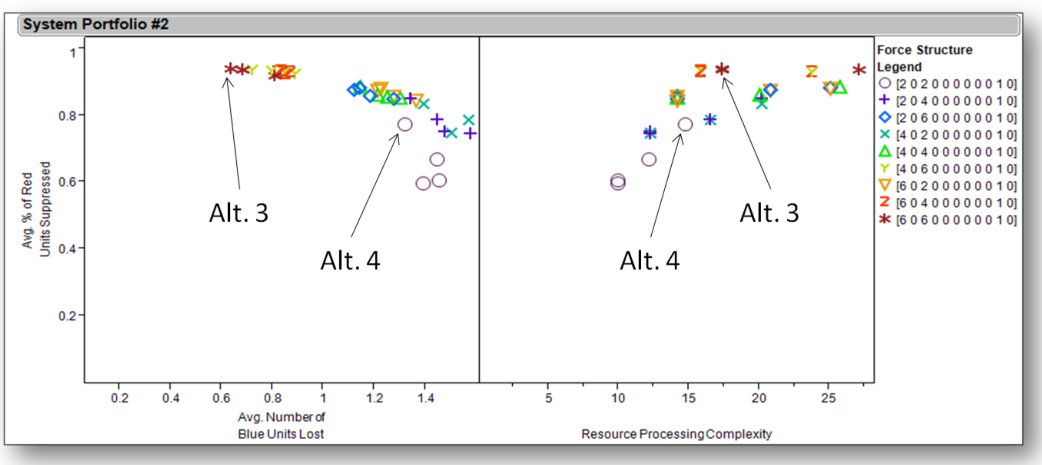


Figure 84: Mission Performance for Alternatives Generated from the 2nd SEAD System Portfolio.

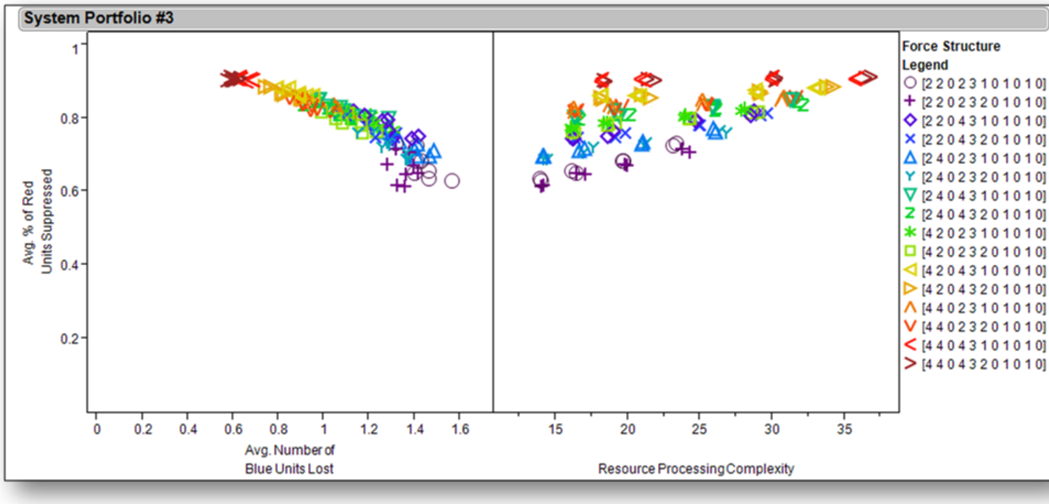


Figure 85: Mission Performance for Alternatives Generated from the 3rd SEAD System Portfolio.

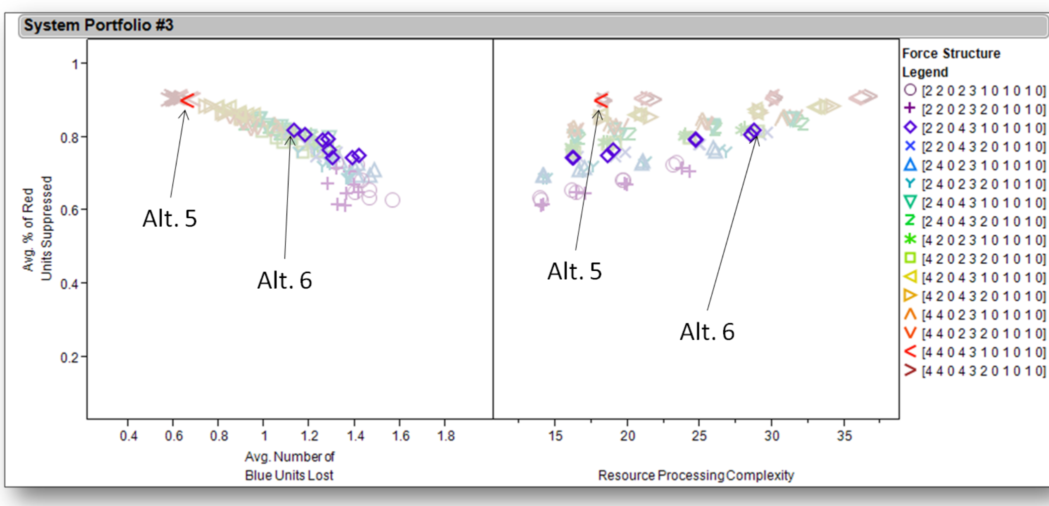


Figure 86: Alternatives Selected from the 3rd SEAD System Portfolio.

For the 3rd SEAD system portfolio, Figure 87 is included to demonstrate the diminishing benefits of collaboration as force size increases. The Packard Commission's 1986 defense acquisition report provides an explanation for the occurrence of this trend [16]:

> At some point, more weapons of lower performance can overcome fewer weapons of higher performance. Hence it is necessary to achieve a critical balance between high military capability and low life cycle cost. In these and other respects, defense acquisition is one of the most difficult management jobs.

An architecture alternative is chosen from each system portfolio grouping that delivers the best performance for a given force structure, collaboration pattern, and IOLs. For comparison, an additional alternative architecture is chosen that is less effective and smaller in terms of force structure. Table 23 provides a summary of the results obtained from Phase I and Phase II M&S for the different alternatives under consideration, as well as the RSC values for each alternative. Additional output data is included in Appendix A.

7.4.5 Life Cycle Cost Analysis

LCC data is obtained from a variety of sources with the goal of obtaining rough order of magnitude purposes for similar systems [2, 86, 87]. A simple estimate of per unit cost for each system is made in order to develop a cost order of magnitude for each alternative. The results are shown in Table 24. The life cycle of each system is assumed to be in the 20–25 year range. Also, in reality, the cost for each system is affected by the amount of missions and capabilities that each system supports, as systems are usually multi-mission in their use. This analysis assumes the same fraction of time-sharing for each system in relation to the time it will be used to conduct SEAD. For a more accurate accounting, the cost for each system can be

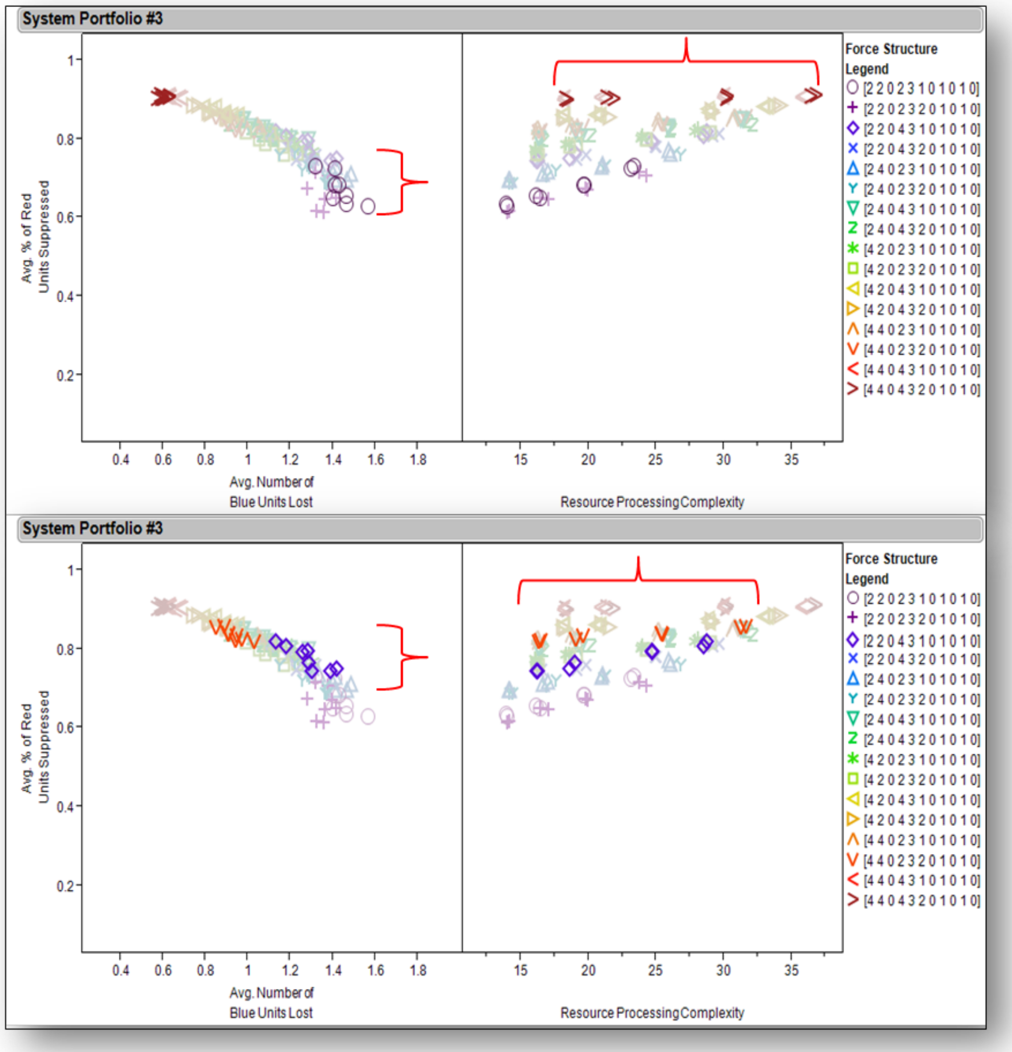


Figure 87: Diminishing Benefits of Collaboration as Force Size Increases for the 3rd SEAD System Portfolio.

Table 23: AOR/JOA Engagement Performance Summary for Blue SEAD SoS Alternatives.

Alt. Number	Portfolio/ ID Number	Included Systems	Force Structure	RPC	RSC	Avg. % of Red Units Suppressed	Standard Deviation	Avg. % of Blue Units Lost	Standard Deviation	Avg. Number of Blue Units Lost
1	1/3.2.2	Central C2 SOF Team Intel Satellite	1 8 1	18.2	1.56	69 %	0.15	11 %	0.13	0.9
2	1/1.1.2	Central C2 SOF Team Intel Satellite	1 5 1	5.4	0.67	48 %	0.15	22 %	0.21	1.1
3	2/8.1.1	CVN F/A-18 X-47B	1 6 6	17.3	0.51	94 %	0.09	5 %	0.09	0.6
4	2/9.2.2	CVN F/A-18 X-47B	1 2 2	14.8	1.54	77 %	0.17	33 %	0.27	1.3
5	3/15.1.1	CVN F/A-18 E-2 EA-6B DDG AH-64 M252	1 4 1 4 1 4 3	18.1	0.67	90 %	0.11	4 %	0.06	0.7
6	3/12.4.2	CVN F/A-18 E-2 EA-6B DDG AH-64 M252	1 2 1 4 1 2 3	28.7	1.70	82 %	0.13	9 %	0.10	1.1

multiplied by an appropriate factor to account for a system's uniqueness in terms of time commitments towards a specific set of capabilities.

Table 24: Estimated Life Cycle Costs.

System Type Type	Est. LCC Per Unit In $Millions
F/A-18	175
AH-64	125
X-47B	200
EA-6B	250
M252	3
DDG	1,700
SOF Team	10
E-2	325
Intel Satellite	3,000
CVN	20,000
Central C2	10,000

7.5 Step 5: Calculate Architecture Complexity & Specify ROA Inputs

The architecture complexity scores for the alternative SEAD architectures are calculated using the following equation:

$$\text{SEAD} \;\; C_\alpha = \frac{\sum_{i=1}^{T} F_i^{\lambda^{(R)}}}{T} \times (1 + \log M) \times \mu\lambda^{(RS)} \times \lambda^{(RP)} \tag{71}$$

The RSC and RPC scores for each alternative are dependent on force structure. These values are contained in Table 23. Based on the aforementioned MoEs, the SEAD R matrix for the SEAD (AOR/JOA) mission consists of two strongly correlated requirements:

$$R_{SEAD} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{72}$$

Using eigenvalue analysis to evaluate the SEAD R matrix, specifically the PFE, $\lambda^{(R)} = 2$. This allows the calculation of FDC for each alternative, based on the systems that are included in each system portfolio. The FDC calculations are given in the following:

$$\left(\frac{\sum_{i=1}^{T} F_i^{\lambda^{(R)}}}{T}\right)_{Alt1} = \left(\frac{10^2 + 6^2 + 5^2}{3}\right) = 53.7 \tag{73}$$

$$\left(\frac{\sum_{i=1}^{T} F_i^{\lambda^{(R)}}}{T}\right)_{Alt2} = \left(\frac{10^2 + 5^2 + 7^2}{3}\right) = 58.0 \tag{74}$$

$$\left(\frac{\sum_{i=1}^{T} F_i^{\lambda^{(R)}}}{T}\right)_{Alt3} = \left(\frac{10^2 + 3 \times (5^2) + 1^2 + 4^2}{6}\right) = 32.0 \tag{75}$$

Using the SEAD system portfolios in conjunction with the information contained in Figure 71, the Cyclomatic Complexity is computed, resulting in the determination of FPC. The results of the calculations are presented in Table 25.

Table 25: SEAD FDC & FPC Scores.

Portfolio Alt. #	FDC	M	FPC $(1 + \log M)$
1	53.7	4	1.60
2	58.0	5	1.70
3	32.0	33	2.52

Now that all four sub-measures have been obtained, the overall architecture complexity score for each alternative can be determined. The results are posted in Table 26. The next step is to scale the architecture complexity scores using Equation (76). Figure 88 provides a graph of the scaling function and plots of both the measured and scaled scores.

$$C'_\alpha(X) = 1 + \left(\frac{1}{10\log_{10}(17)}\right)\log_{10}(X) \tag{76}$$

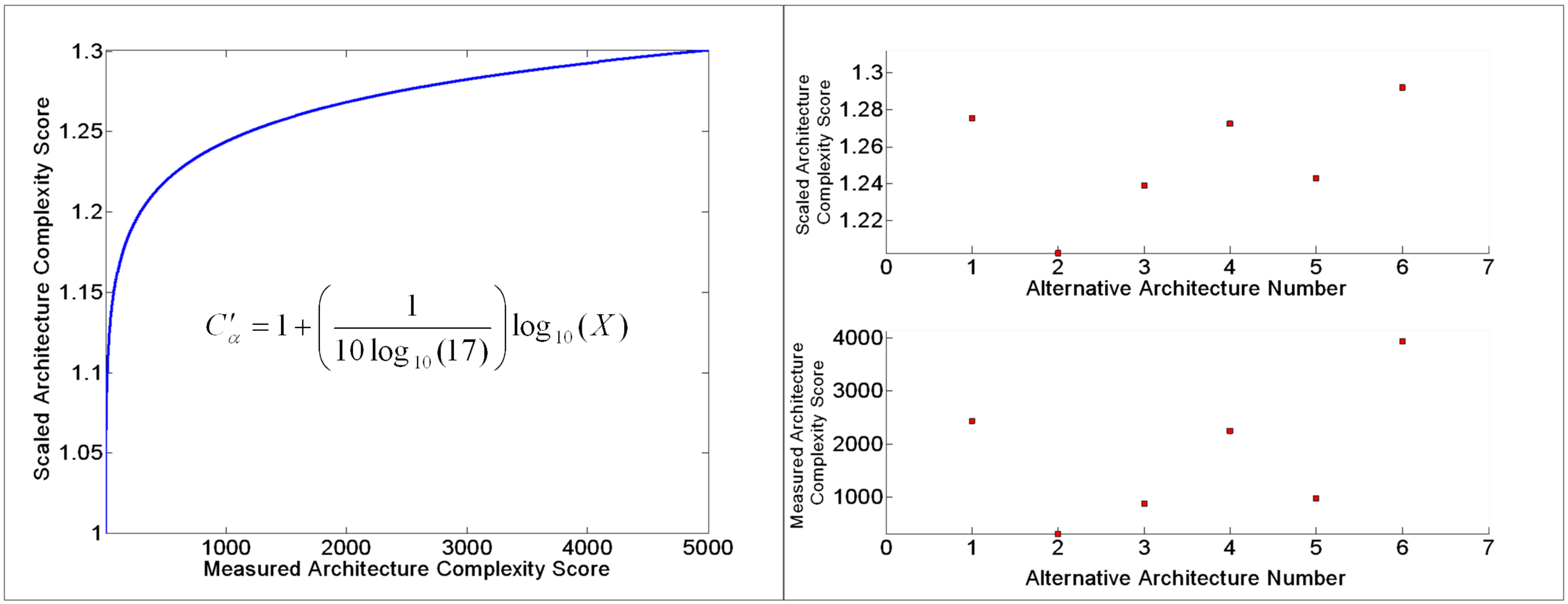


Figure 88: Architecture Complexity Scaling for SEAD Alternatives.

Next, a time to IOC is determined for each alternative based on included system types and force structure. Also, a PoPS is assessed for each alternative by taking into account the types of systems included in the architecture as well as the level of integration required due to IOL requirements. Some systems, such as the X-47B are still in development and make use of advanced technology. Other systems are more readily available and/or face less challenges in their development. The Naval PoPS formulation is also used as a guide in assigning PoPS scores. The following is a brief description of different PoPS categories [23, 66]:

- Green Program (90%–100% PoPS): Program is on track to provide capability, supportablility, and life cycle systems engineering requirements within approved cost and schedule constraints.
- Yellow Program (66%–$<$ 90% PoPS): Program has identified some significant issues with providing capability, supportability, and/or life cycle systems engineerging requirements within approved cost and schedule constraints, but mitigation strategies are being executed.
- Red Program ($<$ 66% PoPS): Program has identified issues that will inhibit delivery of capability, supportability, and/or life cycle systems engineering requirements within approved cost and schedule constraints.

This completes the input requirements for the ARC-VM inputs. Using these inputs, and the results from the M&S acquisition value can be determined. The results are presented in Table 26. Figure 89 provides an accounting of some of the important programmatic parameters contained in Table 26, merely to provide a more visual way of making comparisons of the different alternatives. Figure 89 also contains plots of the utility functions used in the analysis.

Table 26: ARC-VM Data & Calculations for SEAD AOR/JOA Alternatives.

$t_\beta = 5$ Years	Alt. 1	Alt. 2	Alt. 3	Alt. 4	Alt. 5	Alt. 6
E	0.69	0.48	0.94	0.77	0.90	0.82
C_α	2,439	311	870	2,247	977	3,934
C'_α	1.275	1.202	1.239	1.272	1.243	1.292
t_α	3	0.5	5.5	7	3.75	5
τ	1.51	3.30	0.90	0.66	1.29	1
PoPS	0.85	0.98	0.75	0.70	0.76	0.70
ϵ	0.15	0.15	0.09	0.17	0.11	0.13
Δ	0.26	0.11	0.21	0.45	0.21	0.29
$PV(C'_\alpha)$	0.51	0.13	0.75	0.90	0.60	0.76
NAV_q	1.36	3.75	1.25	0.86	1.50	1.07
LCC ($M)	13,080	13,050	22,250	20,750	24,234	23,634

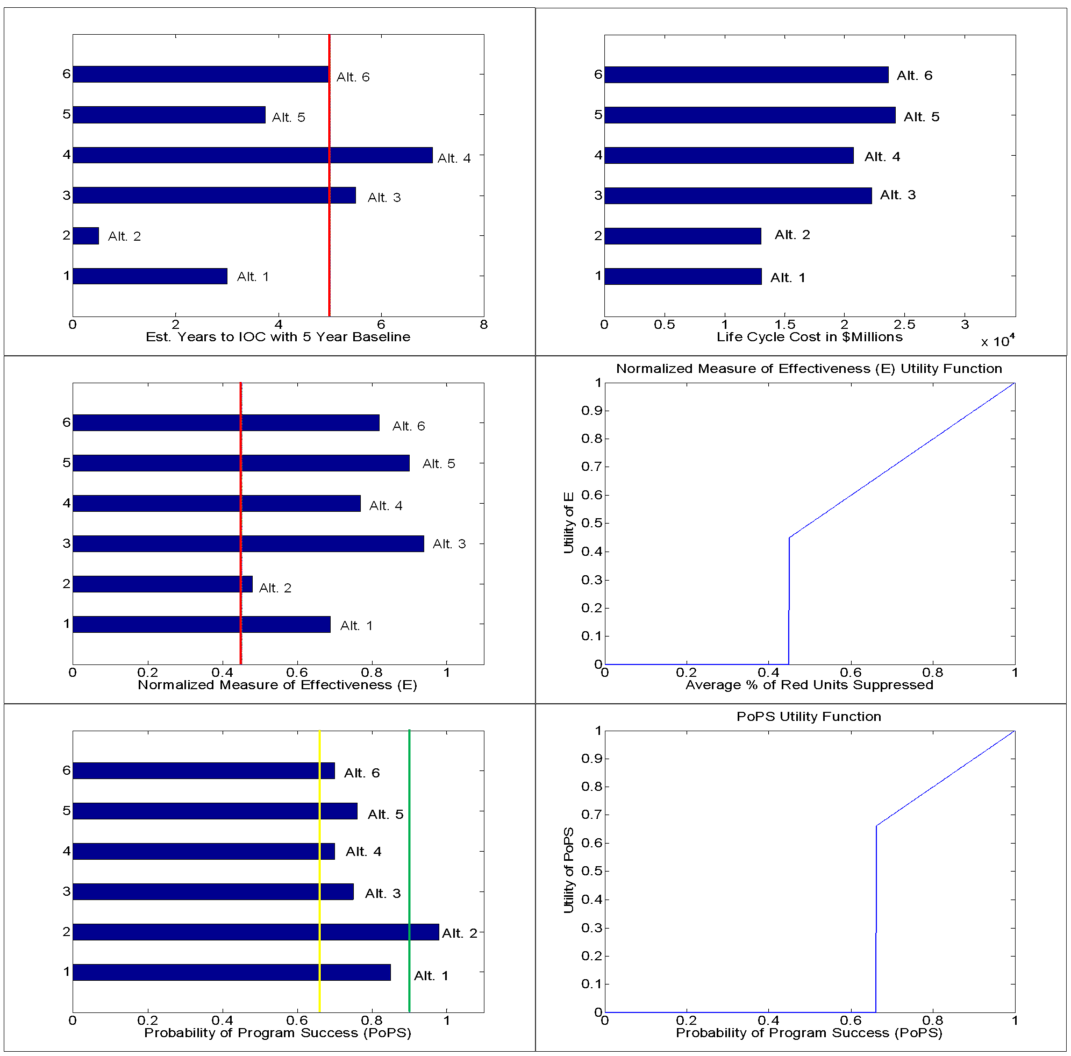


Figure 89: Summary of SEAD Programmatic Parameters.

7.6 Step 6: Conduct Analysis of Alternatives

The AOS and Cost-Effectiveness scatter plot for the SEAD AOR/JOA mission are shown in Figures 90 and 91, respectively. Note that the y-axis has been inverted in order to make use of the plotting software. Figure 92 shows possible trajectories that alternatives may take within the option space. The lefthand AOS is meant to capture the negative effects of schedule slips, where the time to acquisition grows to twice the desired IOC for each alternative. On the other hand, the righthand AOS in Figure 92 shows the position in the AOS when only 6 months are remaining to reach IOC for each alternative. It should be stressed that these plots are only meant to provide a sense of the possible trajectories that the alternatives may take within the AOS. The actual trajectory of an alternative depends on a number of dynamic factors, including changes along the way in PoPS, rated effectiveness, and variance in effectiveness.

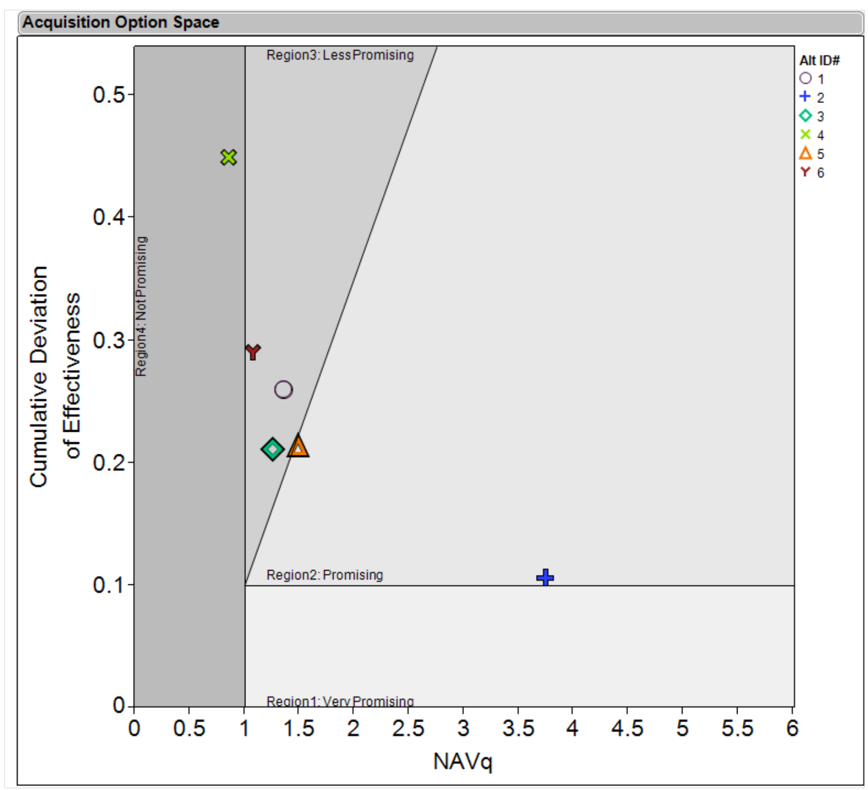


Figure 90: SEAD AOR/JOA AOS ($E_B = 0.75$, $\epsilon_B = 0.1$).

Figures 93 and 94 help illustrate the effects of changing requirements on the size

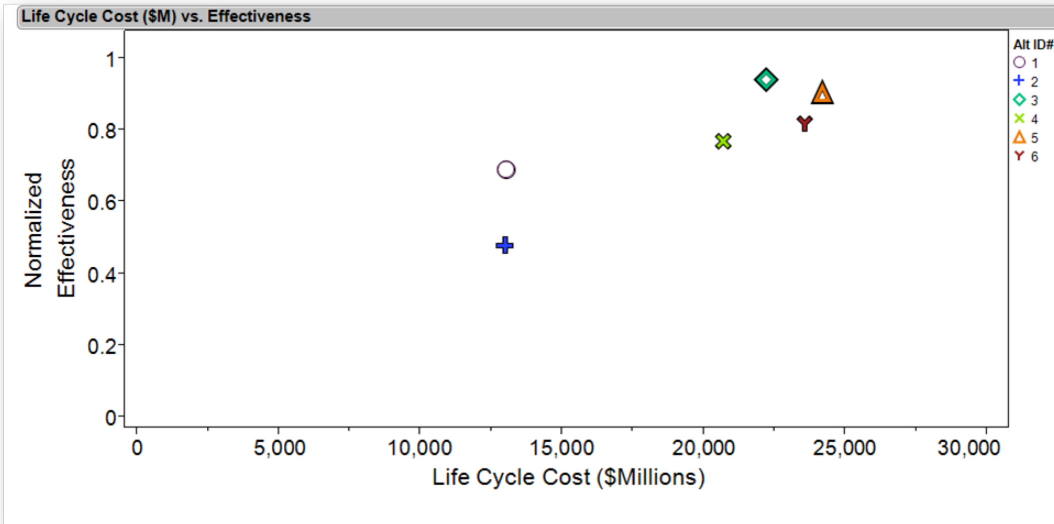


Figure 91: SEAD AOR/JOA Cost-Effectiveness Scatter Plot.

and shape of the AOS regions. Figure 93 shows the effects of changing the required baseline effectiveness. $E_B = 0.60$ for the lefthand AOS in the figure and $E_B = 0.95$ for the righthand figure, while $\epsilon_B = 0.1$ for both. As greater effectiveness is required for conducting SEAD, region 2 gives way to region 3, which grows in area. In Figure 94, a change in the variance of the required effectiveness causes region 1 to change in size. In the lefthand figure $\epsilon_B = 0.05$ and $\epsilon_B = 0.2$ in righthand figure. For both plots in Figure 94 $E_B = 0.75$. Thus, if the SoS is expected to conduct SEAD with very little variation in performance, the AOS reflects that the alternatives have a farther distance to traverse to reach region 1.

From this analysis, it is clear to see that Alternatives 2, 3, and 5 are the most likely candidates for acquisition. While Alt. 2 provides low initial capability, that capability can be fielding quickly at low cost, with minimum risk, and limited complexity. If decision makers agree that Alt. 2 is not worth pursuing, then the threshold value can be raised to lower the NAVq of Alt. 2 and update the AOS to reflect the decision makers' shifting preferences. However, it is worth considering that Alt. 2 could become part of an evolutionary acquisition strategy. This strategy could encompass

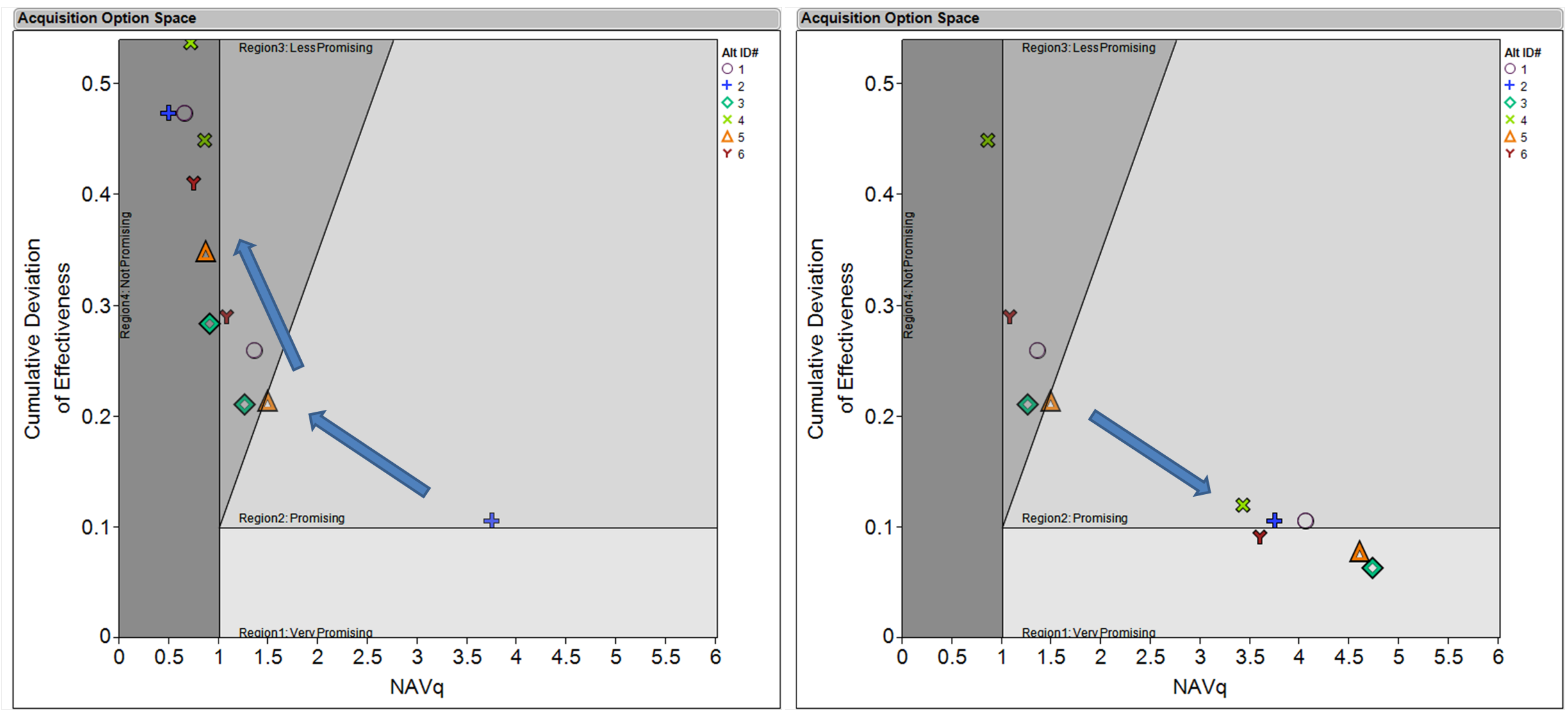


Figure 92: Possible Trajectories for the Alternatives in the SEAD AOR/JOA Acquisition Option Space ($E_B = 0.75$, $\epsilon_B = 0.1$).

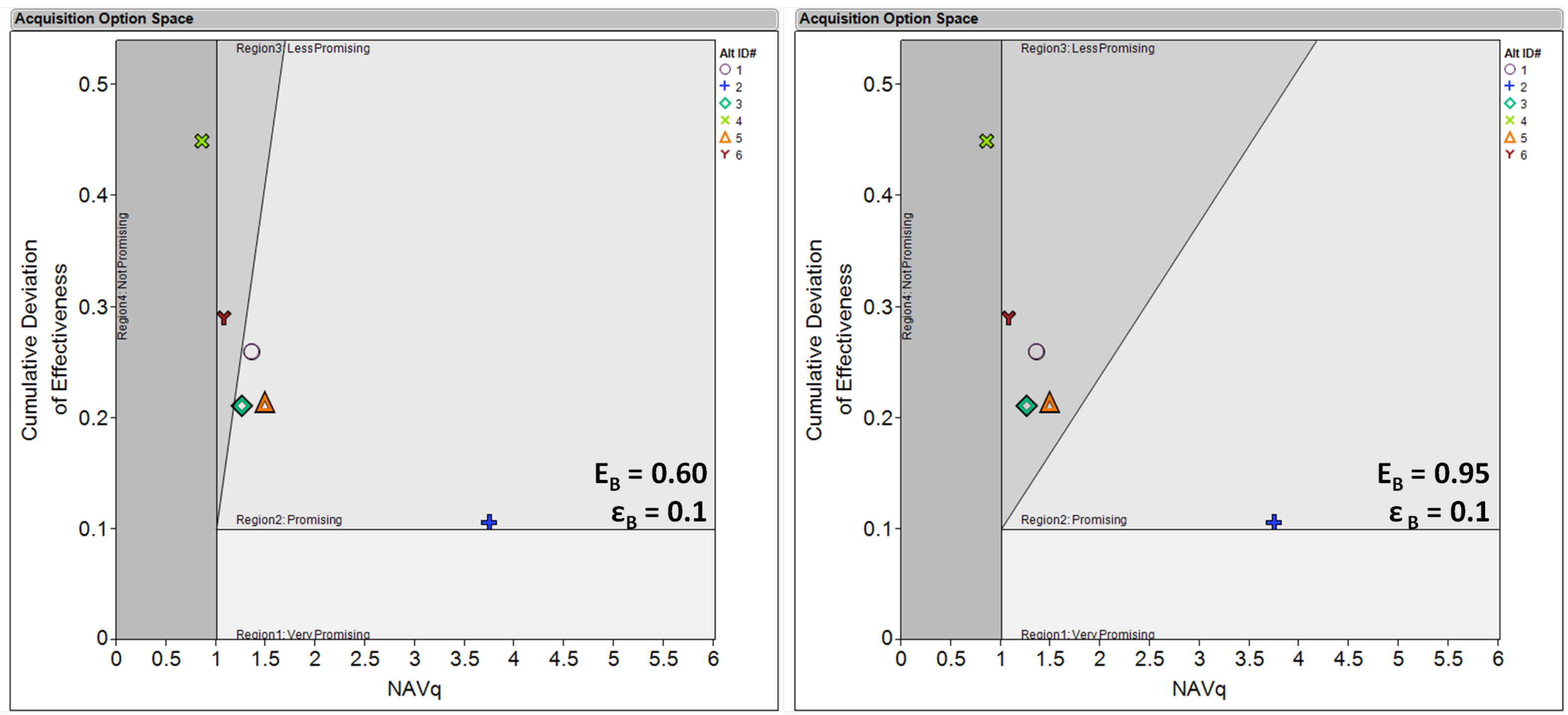


Figure 93: Change in AOS Regions with Changes In Required Baseline Effectiveness.

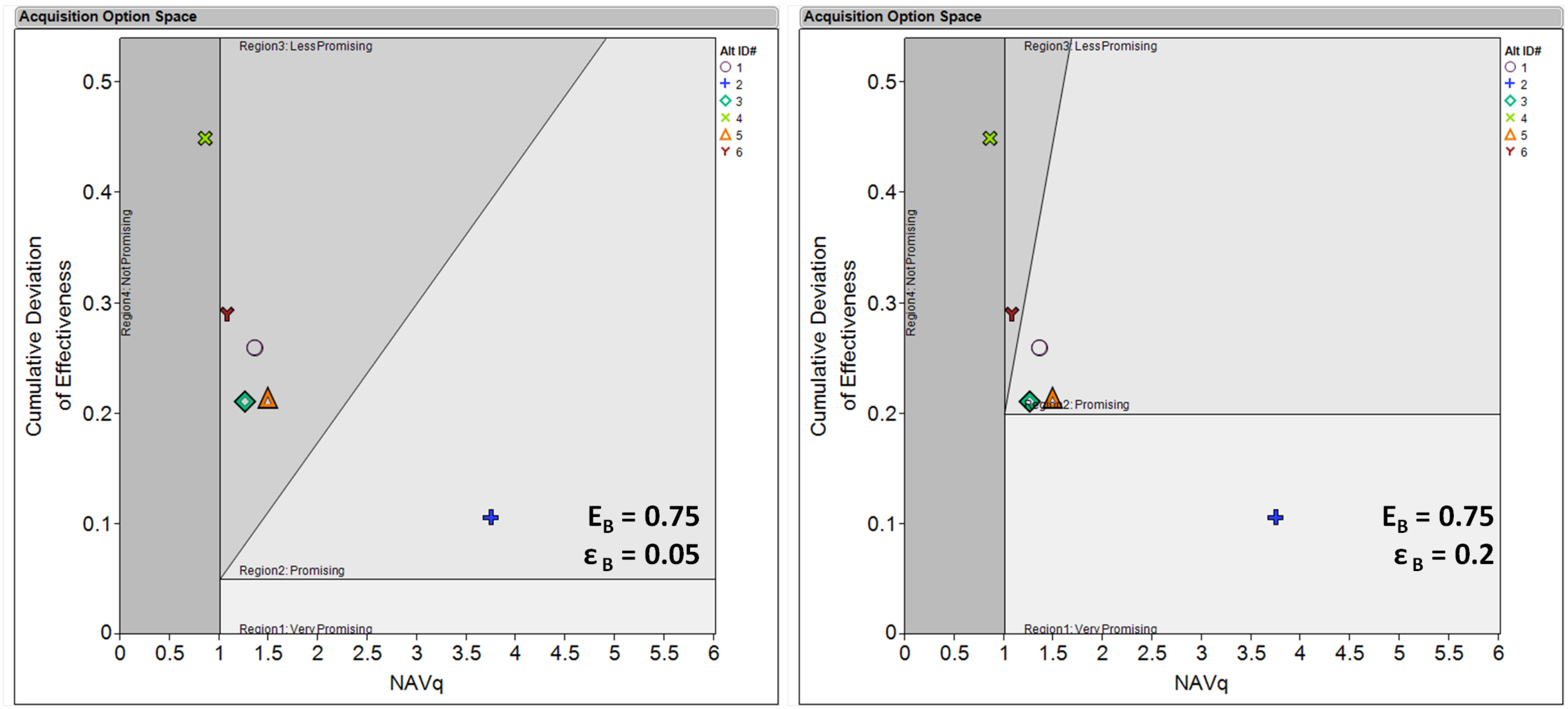


Figure 94: Change in AOS Regions with Changes In Required Baseline Effectiveness Variance.

quickly fielding Alt. 2 while continuing to simultaneously develop Alt. 1, which is compromised of the same systems but is expected to take longer to reach IOC due to a more complex architecture that relies more heavily on networking.

Based on its position in the option space, Alt. 5 is examined next. Though Alt. 5 is relatively complex and costly, it delivers a high enough return on capability to offset these costs. While the programmatic risk for Alt. 5 could be described as merely acceptable in comparison to some of the other alternatives, the relatively quick time to achieve IOC makes this alternative more attractive. Thus, while Alt. 5 is the costliest of all the alternatives, it still represents a valuable opportunity to pursue, budget withstanding. If a lower cost solution is desired due to budget constraints, then Alt. 1, with a $NAVq = 1.36$ and PoPS of 85%, also proves to be a valuable solution. While less effective than some of the other alternatives, it is a relatively low cost, low complexity alternative. With that said, the AOS clearly shows that additional oversight may still be needed. This is primarily due to the relatively higher variance of effectiveness, which contributes significantly to the overall cumulative deviation of effectiveness for this alternative. Any moderate increases in risk or schedule will jeapordize any value of Alt. 1 and may be cause for program termination.

Meanwhile, Alt. 3 represents the best performing alternative. Alt. 3 is compromised of F/A-18's and X-47B and is an example of an architecture where autonomous UAV's are working alongside manned aerial components to conduct SEAD operations. Though more costly in term of budget, primarily due to the new technologies that must be developed, the AOS shows that expending resources to pursue this alternative holds value. The AOS also recommends, however, that additional oversight be afforded to keep this alternative from entering Region 4 due to the complexities involved. The AOS does show, however, that the PoPS would have to drop to approximately 40% before this alternative would cease to hold acquisition value.

Lastly, the AOS is able to confirm the poor value of alternatives that should not be recommended. Alt. 6 for example, is an architecture with poor performance, high complexity, low PoPS, and will take approximately two years longer than desired to be fielded. Even if this alternative were to be considered low cost in terms of budget, it would not represent a good value by any means. Thus, by providing a means for assessment independent of monetary costs, the AOS has proven useful in determining when additional performance is worth the investment of additional resources. Using the SEAD example, it has been shown that the AOS can help to determine the value of increased levels of interoperability and resource sharing when comparing alternatives. Use of the AOS also provides decision makers with the capability to quantify how changes in the PoPS can impact the value of an acquisition. It was also demonstrated how the AOS can be used to help craft evolutionary acquisition strategies. It should also be noted that this analysis assumes a neutral risk tolerance. Inclusion of risk tolerances will affect the recommendations provided by the AOS analysis. In all, the AOS proves to be a powerful analysis tool for aiding acquisition-level decision making.

CHAPTER VIII

SUMMARY & CONCLUSIONS

The purpose of this research was to document the development of a methodology designed to address some of the shortcomings that have come to plague military acquisition, namely chronic cost and schedule overruns in the pursuit of advanced weaponry, tactics, and operations. The problem is compounded by the necessity for the acquisition of Systems-of-Systems, which themselves can be classified as complex systems. Thus, while Systems-of-Systems bring added capabilities to help achieve greater levels of military effectiveness, they also introduce additional complexities in fielding and maintaining new weapons systems critical to national defense. This led to the formulation of the research objective, stated in the following:

> **Research Objective: Allow for more informed tradeoffs between cost, schedule, and performance during the pre-Milestone A phase of military SoS acquisitions by developing a valuation methodology suitable for acquisition-level decision making. The valuation methodology should provide a measure of System-of-System architecture complexity and a conceptual and visual framework to quantifiably and traceably combine complexity, time-valued performance levels, programmatic risks, and the uncertainties in performance and schedule estimates.**

First, a framework was developed in Chapter 5 to measure the architecture complexity of a System-of-System. To accomplish this, it was necessary to identify precise definitions of both an *architecture* and a *complex system* in the context of military SoS acquisition. This made it possible to apply research from the field of Complexity

Science in order to determine which of the various approaches to measuring complexity would prove useful in developing an architecture complexity measure, with the result being that measuring the degree of organization of a system would be the most sensible approach. Next, criteria were identified that would facilitate judging the overall applicability of the developed architecture complexity measure. A survey of existing system complexity measures was conducted, and while no single existing measure proved to be directly applicable, each of the existing measures provided important insights as to what features should be captured in providing a comprehensive accounting of architecture complexity. Alternative methods were identified to the formulation of each sub-measure, allowing for flexibility in accounting for problem-specific nuances that may arise.

With this knowledge in hand, individual sub-measures were identified, each designed to capture an important observable feature of the complex system architecture. The application of measurement theory and utility theory aided in the development of a logical framework to combine the independent sub-measures into an overall architecture complexity score. Additionally, the inclusion of graph theory directly resulted in an automatable, objective, and mathematically rigorous measure of architecture complexity that met all of the applicability criteria. The primary benefit to system architects is the ability to compare the complexity of different architectures through careful analysis of the arrangement of functionality and resource sharing. Specifically, the framework addresses the tradeoffs that occur when component reduction is achieved at the expense of fewer, more complex systems. It also addresses the additional complexity that results from overlapping functionality as multiple components must interact in a programmed sequence. The impact on complexity from increased interoperability between components is also included. Lastly, the factors that affect the underlying patterns of resource exchange is another important aspect to consider in selecting a military system-of-system architecture.

The need for assessing the value of allocating resources towards a particular acquisition is not unique to defense acquisition. However, the fundamental structure of the customer-supplier relationship makes it difficult to assess a fair market price for the development and purchase of military systems. Because complexity can be considered as the cost inherent in a design for achieving increased functionality, efficiency, or flexibility, this research identified that making the conceptual shift of "monetizing" complexity, or treating complexity as a cost, allows for more advanced financial theories and methods to be used. This resulted in the use of Real Options, and Chapter 6 details the development of an Acquisition Option Space as a way to provide decision makers with a conceptual and visual framework with which to value and compare alternatives.

The Acquisition Option Space supports the research objective by providing an analysis framework for taking into account programmatic risks, uncertainties, and sensitivities while also pushing to the forefront the hidden costs of complexity and emphasizing time-valued capability. The AOS is a powerful decision aid, structured in such a way that it provides understandable interpretations using an intuitive visualization format. This helps to ensure that the advantages and disadvantages of each alternative are presented in a clear and unbiased manner. Finally, development of the Acquisition Option Space allows for the creation of the overall valuation methodology. ARC-VM provides a means for acquisition-level decision makers to quantify complexity, which is one of the main drivers of unsuccessful acquisition attempts, and to use this measured complexity as the basis in determining acquisition value.

ARC-VM begins with clearly defining capability requirements, since they have a profound impact on system complexity and acquisition program success. The next step is to define alternative system portfolios that define which systems will be grouped together as a SoS. This makes it possible to move on the next step of generating feasible alternative architectures that can vary in force structure, patterns of collaboration, interoperability levels, functional process sequencing, and technology. In order to assess the performance of each alternative architecture, a modeling and simulation environment was created, using a suppression of enemy air defenses scenario as a test case. The modeling and simulation environment includes an architecture resource-based collaborative network evaluation tool to estimate the benefits of increased knowledge that results from systems collaborating as part of a system-of-systems. In conjunction with the performance assessments, a notional cost analysis was conducted for later use in the analysis of alternatives.

Prior to the analysis of alternatives, the complexity score for each alternative architecture is calculated. With this information in hand, the analysis of alternatives can be conducted using the Acquisition Option Space, cost-effectiveness scatter plot, and any necessary summary portfolio views. The end result of applying the ARC-VM methodology is an assessment of acquisition value—that is, an assessment of when additional effectiveness is worth an additional investment of resources when comparing alternatives. Because the Acquisition Option Space does not use monetary costs in its formulation of acquisition value, ARC-VM provides an important alternative analysis that can be used in conjunction with existing techniques. This provides decision makers with a better overall assessment of alternatives, which is the first step in developing an evolutionary acquisition strategy to provide warfighters with timely, affordable military capabilities.

8.1 Recommendations for Future Work

A key element of performing a successful analysis of alternatives is ensuring that threats and scenarios are realistic, and that a broad range of environmental and hostile operating environments are considered. The identification of threat and scenario aspects deemed most influential to the outcome of the analysis should also be performed [127]. While the scenario developed for this study addresses some of these issues, for example by including the uncertainty that may be faced due to varying enemy force structure and location, further work is needed. This work should also include studying the impact of new technologies on architecture development. Accomplishing this requires the integration of additional M&S environments into the overall analysis. Also of particular note, further research into the subject area of interoperability is needed. A methodology for more accurately determining the shape of the IOL utility curves used in this research would be of great benefit, especially if it can be determined how the IOL utility curves change in relation to each other and under what conditions and assumptions.

For purposes of this research, a simplified M&S environment was created to model the engagements between Blue and Red forces. The aim of this research was to provide a proof of concept for the overall ARC-VM methodology. In the future, to adequately assess all aspects of architecture performance, more sophisticated and diverse M&S environments should be used to provide a more robust measure of all aspects of architecture performance. For example, the Red IADS itself should eventually be modeled as a complex SoS as well during simulated engagements, with its own emergent and adaptive behavior. The principal benefit that arises from the development of ARC-VM is in providing an overarching framework to combine M&S results with other information necessary from an acquisition standpoint to provide decision makers with a comprehensive assessment of value.

In reality, military systems and SoS must provide capabilities across broad mission areas. For instance, the suppression of enemy air defenses capability encompasses multiple competing mission areas in addition to the AOR/JOA mission studied here. Additional mission areas include localized suppression and opportune suppression. The inclusion of these additional mission areas in the analysis may require capability tradeoffs and evolutionary acquisition to fully satisfy all of the mission area requirements. In addition, the opportunity exists to investigate and quantify the correlation between added complexity and system flexibility, adaptability, and robustness. A key question that may be addressed is how much added complexity is usually needed for a system to realize significant improvements in any of the aforementioned traits, and under what circumstances.

This research represents a preliminary attempt at incorporating complexity-based Real Options into the acquisition decision-making process. The use of Real Options should be investigated further in order to determine the feasibility of applying more advanced Real Options techniques. For example, the use of nested options may prove useful, particularly in more fully developing evolutionary acquisition strategies. As with any newly developed methodology, validation requires the use of the methodology in real-world applications to a variety of test cases. This will provide a basis for comparing measured architecture complexity scores against the outcomes of actual acquisition programs. Also, the strength of the relationship between the architecture complexity measure developed in this research and developed cost estimating relationships should be explored. One area of research would be to investigate if the architecture complexity framework developed in this thesis can lead to better cost estimating relationships and reduce the relatively large uncertainty in cost estimates seen during the initial stages of design.

APPENDIX A

SEAD M&S OUTPUT DATA

Table 27: Output Data for Portfolio 1 Alternatives.

ID.	FS.	Collab Matrix.	IOL RefTable.	FS	RP List Index	RPC	Num_ Blue Lost Mean	Blue % Lost Mean	Blue % Lost Std Dev	Red % Sup_ Mean	Red % Sup_ Std Dev	RP.
1	1	1	1	[0 0 0 0 0 0 5 0 1 0 1]	1, 1, 1	5.385164807	1.014444444	0.202888889	0.219496058	0.486039242	0.153642992	1
1	1	2	1	[0 0 0 0 0 0 5 0 1 0 1]	1, 1, 2	10.99345706	0.994444444	0.198888889	0.220725089	0.610484568	0.16719215	2
1	1	1	2	[0 0 0 0 0 0 5 0 1 0 1]	1, 1, 3	5.385164807	1.073333333	0.214666667	0.219531304	0.477640212	0.146637461	3
1	1	2	2	[0 0 0 0 0 0 5 0 1 0 1]	1, 1, 4	12.30073525	0.97	0.194	0.215339451	0.650511464	0.163389403	4
1	2	1	1	[0 0 0 0 0 0 10 0 1 0 1]	1, 2, 1	7	0.877777778	0.087777778	0.105074557	0.48593739	0.133897861	1
1	2	2	1	[0 0 0 0 0 0 10 0 1 0 1]	1, 2, 2	20.7462991	0.856666667	0.085666667	0.107064689	0.661138007	0.152564127	2
1	2	1	2	[0 0 0 0 0 0 10 0 1 0 1]	1, 2, 3	7	0.973333333	0.097333333	0.102978987	0.48358642	0.127641322	3
1	2	2	2	[0 0 0 0 0 0 10 0 1 0 1]	1, 2, 4	22.17261753	0.888888889	0.088888889	0.106876416	0.673349206	0.157167113	4
1	3	1	1	[0 0 0 0 0 0 8 0 1 0 1]	1, 3, 1	6.403124237	0.947777778	0.118472222	0.142634416	0.494808642	0.135628119	1
1	3	2	1	[0 0 0 0 0 0 8 0 1 0 1]	1, 3, 2	16.80893649	0.888888889	0.111111111	0.135276213	0.652578483	0.156399506	2
1	3	1	2	[0 0 0 0 0 0 8 0 1 0 1]	1, 3, 3	6.403124237	0.993333333	0.124166667	0.145015917	0.491415344	0.138485613	3
1	3	2	2	[0 0 0 0 0 0 8 0 1 0 1]	1, 3, 4	18.20824392	0.901111111	0.112638889	0.133420424	0.686840829	0.150252409	4

Table 28: Output Data for Portfolio 2 Alternatives.

ID.	FS.	Collab Matrix.	IOL RefTable.	FS	RP List Index	RPC	Num. Blue Lost Mean	Blue % Lost Mean	Blue % Lost Std Dev	Red % Sup. Mean	Red % Sup. Std Dev	RP.
2	1	1	1	[4 0 4 0 0 0 0 0 0 1 0]	1, 1, 1	14.14213562	1.3	0.1625	0.174151766	0.853039683	0.128052645	1
2	1	2	1	[4 0 4 0 0 0 0 0 0 1 0]	1, 1, 2	20	1.214444444	0.151805556	0.176855933	0.862853616	0.12660484	2
2	1	1	2	[4 0 4 0 0 0 0 0 0 1 0]	1, 1, 3	14.14213562	1.248888889	0.156111111	0.166459363	0.855799824	0.126423663	3
2	1	2	2	[4 0 4 0 0 0 0 0 0 1 0]	1, 1, 4	25.76305461	1.146666667	0.143333333	0.164484228	0.885143739	0.122211721	4
2	2	1	1	[2 0 6 0 0 0 0 0 0 1 0]	1, 2, 1	14.14213562	1.275555556	0.159444444	0.172177613	0.850590388	0.122127199	1
2	2	2	1	[2 0 6 0 0 0 0 0 0 1 0]	1, 2, 2	20.77748957	1.122222222	0.140277778	0.160738806	0.875364638	0.120262128	2
2	2	1	2	[2 0 6 0 0 0 0 0 0 1 0]	1, 2, 3	14.14213562	1.182222222	0.147777778	0.163715058	0.857608907	0.129013512	3
2	2	2	2	[2 0 6 0 0 0 0 0 0 1 0]	1, 2, 4	25.07562238	1.142222222	0.142777778	0.170663514	0.881770282	0.11932337	4
2	3	1	1	[2 0 4 0 0 0 0 0 0 1 0]	1, 3, 1	12.24744871	1.576666667	0.262777778	0.231449971	0.743449735	0.153172153	1
2	3	2	1	[2 0 4 0 0 0 0 0 0 1 0]	1, 3, 2	16.48738389	1.446666667	0.241111111	0.227187454	0.787272487	0.155759155	2
2	3	1	2	[2 0 4 0 0 0 0 0 0 1 0]	1, 3, 3	12.24744871	1.478888889	0.246481481	0.223634356	0.751009259	0.156890637	3
2	3	2	2	[2 0 4 0 0 0 0 0 0 1 0]	1, 3, 4	20.14755904	1.342222222	0.223703704	0.220573541	0.848331129	0.136704208	4
2	4	1	1	[6 0 4 0 0 0 0 0 0 1 0]	1, 4, 1	15.8113883	0.833333333	0.083333333	0.118246363	0.933878748	0.088696431	1
2	4	2	1	[6 0 4 0 0 0 0 0 0 1 0]	1, 4, 2	23.76532003	0.868888889	0.086888889	0.120368774	0.930091711	0.092276042	2
2	4	1	2	[6 0 4 0 0 0 0 0 0 1 0]	1, 4, 3	15.8113883	0.823333333	0.082333333	0.11824213	0.931582451	0.089484277	3
2	4	2	2	[6 0 4 0 0 0 0 0 0 1 0]	1, 4, 4	30.91273692	0.851111111	0.085111111	0.12062356	0.925121693	0.095576368	4
2	5	1	1	[4 0 2 0 0 0 0 0 0 1 0]	1, 5, 1	12.24744871	1.503333333	0.250555556	0.225313059	0.746615961	0.157896214	1
2	5	2	1	[4 0 2 0 0 0 0 0 0 1 0]	1, 5, 2	16.48738389	1.57	0.261666667	0.236917621	0.788186067	0.151387635	2
2	5	1	2	[4 0 2 0 0 0 0 0 0 1 0]	1, 5, 3	12.24744871	1.664444444	0.277407407	0.230577388	0.743512346	0.157330075	3
2	5	2	2	[4 0 2 0 0 0 0 0 0 1 0]	1, 5, 4	20.14755904	1.394444444	0.232407407	0.223245312	0.835136684	0.139461126	4
2	6	1	1	[6 0 2 0 0 0 0 0 0 1 0]	1, 6, 1	14.14213562	1.277777778	0.159722222	0.171413448	0.855184303	0.130001691	1
2	6	2	1	[6 0 2 0 0 0 0 0 0 1 0]	1, 6, 2	20.77748957	1.211111111	0.151388889	0.166753546	0.874388448	0.119408429	2
2	6	1	2	[6 0 2 0 0 0 0 0 0 1 0]	1, 6, 3	14.14213562	1.362222222	0.170277778	0.179891299	0.843889771	0.128784238	3
2	6	2	2	[6 0 2 0 0 0 0 0 0 1 0]	1, 6, 4	25.07562238	1.224444444	0.153055556	0.171418856	0.879986772	0.115205339	4
2	7	1	1	[4 0 6 0 0 0 0 0 0 1 0]	1, 7, 1	15.8113883	0.721111111	0.072111111	0.110766311	0.936354938	0.088787083	1
2	7	2	1	[4 0 6 0 0 0 0 0 0 1 0]	1, 7, 2	23.76532003	0.848888889	0.084888889	0.118830456	0.931313933	0.096224445	2
2	7	1	2	[4 0 6 0 0 0 0 0 0 1 0]	1, 7, 3	15.8113883	0.796666667	0.079666667	0.116387504	0.933138448	0.093717468	3
2	7	2	2	[4 0 6 0 0 0 0 0 0 1 0]	1, 7, 4	30.91273692	0.891111111	0.089111111	0.124135672	0.923047178	0.095043189	4
2	8	1	1	[6 0 6 0 0 0 0 0 0 1 0]	1, 8, 1	17.32050808	0.638888889	0.053240741	0.087256507	0.940424603	0.085609342	1
2	8	2	1	[6 0 6 0 0 0 0 0 0 1 0]	1, 8, 2	27.07878403	0.683333333	0.056944444	0.091622836	0.936935626	0.087871064	2
2	8	1	2	[6 0 6 0 0 0 0 0 0 1 0]	1, 8, 3	17.32050808	0.682222222	0.056851852	0.091722501	0.935125661	0.089061258	3
2	8	2	2	[6 0 6 0 0 0 0 0 0 1 0]	1, 8, 4	36.27105745	0.807777778	0.067314815	0.100200082	0.919681217	0.100909402	4
2	9	1	1	[2 0 2 0 0 0 0 0 0 1 0]	1, 9, 1	10	1.455555556	0.363888889	0.283640698	0.604774691	0.184915659	1
2	9	2	1	[2 0 2 0 0 0 0 0 0 1 0]	1, 9, 2	12.19803903	1.448888889	0.362222222	0.279868783	0.667535714	0.177130355	2
2	9	1	2	[2 0 2 0 0 0 0 0 0 1 0]	1, 9, 3	10	1.392222222	0.348055556	0.281836298	0.595101411	0.178177384	3
2	9	2	2	[2 0 2 0 0 0 0 0 0 1 0]	1, 9, 4	14.77032961	1.322222222	0.330555556	0.27383025	0.772810406	0.168321814	4

Table 29: Output Data for Portfolio 3 Alternatives.

ID.	FS.	Collab Matrix.	IOL RefTable.	FS	RP List Index	RPC	Num_ Blue Lost Mean	Blue % Lost Mean	Blue % Lost Std Dev	Red % Sup_ Mean	Red % Sup_ Std Dev	RP.
3	1	1	1	[2 2 0 4 3 2 0 1 0 1 0]	1, 1, 1	16.32169907	1.291111111	0.092222222	0.090392517	0.744500441	0.143093551	1
3	1	2	1	[2 2 0 4 3 2 0 1 0 1 0]	1, 1, 2	24.87125407	1.184444444	0.084603175	0.091504909	0.780525132	0.140340227	2
3	1	3	1	[2 2 0 4 3 2 0 1 0 1 0]	1, 1, 3	16.51179381	1.227777778	0.087698413	0.089527499	0.749373898	0.149873875	3
3	1	4	1	[2 2 0 4 3 2 0 1 0 1 0]	1, 1, 4	24.99414138	1.186666667	0.084761905	0.08584925	0.782195326	0.13448366	4
3	1	1	2	[2 2 0 4 3 2 0 1 0 1 0]	1, 1, 5	19.13939549	1.308888889	0.093492063	0.091092988	0.748770282	0.142864055	5
3	1	2	2	[2 2 0 4 3 2 0 1 0 1 0]	1, 1, 6	29.1887437	1.097777778	0.078412698	0.085675727	0.808163139	0.137169495	6
3	1	3	2	[2 2 0 4 3 2 0 1 0 1 0]	1, 1, 7	19.81170375	1.317777778	0.094126984	0.091620608	0.760891534	0.141601508	7
3	1	4	2	[2 2 0 4 3 2 0 1 0 1 0]	1, 1, 8	29.63217156	1.066666667	0.076190476	0.085828093	0.813737654	0.133238339	8
3	2	1	1	[2 2 0 2 3 1 0 1 0 1 0]	1, 2, 1	13.90380081	1.463333333	0.133030303	0.120739759	0.632838624	0.155548042	1
3	2	2	1	[2 2 0 2 3 1 0 1 0 1 0]	1, 2, 2	19.61434103	1.41	0.128181818	0.119127034	0.682234568	0.15098436	2
3	2	3	1	[2 2 0 2 3 1 0 1 0 1 0]	1, 2, 3	13.99803623	1.567777778	0.142525253	0.118242303	0.629245591	0.151634834	3
3	2	4	1	[2 2 0 2 3 1 0 1 0 1 0]	1, 2, 4	19.68360905	1.432222222	0.13020202	0.127013357	0.681733686	0.153237701	4
3	2	1	2	[2 2 0 2 3 1 0 1 0 1 0]	1, 2, 5	16.10390977	1.465555556	0.133232323	0.120745174	0.653859347	0.154411192	5
3	2	2	2	[2 2 0 2 3 1 0 1 0 1 0]	1, 2, 6	23.15492635	1.413333333	0.128484848	0.123019524	0.724059524	0.147327048	6
3	2	3	2	[2 2 0 2 3 1 0 1 0 1 0]	1, 2, 7	16.4419071	1.404444444	0.127676768	0.117223971	0.649990741	0.153149312	7
3	2	4	2	[2 2 0 2 3 1 0 1 0 1 0]	1, 2, 8	23.40198596	1.321111111	0.12010101	0.121589857	0.730216931	0.149972661	8
3	3	1	1	[2 4 0 2 3 2 0 1 0 1 0]	1, 3, 1	14.17473254	1.362222222	0.097301587	0.090711735	0.687196208	0.153783713	1
3	3	2	1	[2 4 0 2 3 2 0 1 0 1 0]	1, 3, 2	21.08910325	1.292222222	0.092301587	0.093187678	0.726824956	0.144994026	2
3	3	3	1	[2 4 0 2 3 2 0 1 0 1 0]	1, 3, 3	14.35733475	1.372222222	0.098015873	0.089018926	0.689074074	0.150010568	3
3	3	4	1	[2 4 0 2 3 2 0 1 0 1 0]	1, 3, 4	21.21735474	1.278888889	0.091349206	0.089739397	0.73263933	0.149286657	4

Table 30: Output Data for Portfolio 3 Alternatives (continued).

ID.	FS.	Collab Matrix.	IOL RefTable.	FS	RP List Index	RPC	Num. Blue Lost Mean	Blue % Lost Mean	Blue % Lost Std Dev	Red % Sup. Mean	Red % Sup. Std Dev	RP.
3	3	1	2	[2 4 0 2 3 2 0 1 0 1 0]	1, 3, 5	16.93063086	1.361111111	0.097222222	0.095518253	0.707149912	0.153432682	5
3	3	2	2	[2 4 0 2 3 2 0 1 0 1 0]	1, 3, 6	26.33721776	1.224444444	0.087460317	0.08670511	0.76633157	0.146960141	6
3	3	3	2	[2 4 0 2 3 2 0 1 0 1 0]	1, 3, 7	17.56413166	1.254444444	0.089603175	0.087424417	0.719684744	0.148831224	7
3	3	4	2	[2 4 0 2 3 2 0 1 0 1 0]	1, 3, 8	26.7642558	1.147777778	0.081984127	0.085806454	0.759566578	0.144148963	8
3	4	1	1	[2 4 0 4 3 1 0 1 0 1 0]	1, 4, 1	16.29153722	1.2	0.08	0.085306633	0.802108907	0.131450795	1
3	4	2	1	[2 4 0 4 3 1 0 1 0 1 0]	1, 4, 2	25.81366283	1.006666667	0.067111111	0.08337107	0.832593474	0.131053362	2
3	4	3	1	[2 4 0 4 3 1 0 1 0 1 0]	1, 4, 3	16.38793822	1.286666667	0.085777778	0.089769339	0.799147707	0.136913898	3
3	4	4	1	[2 4 0 4 3 1 0 1 0 1 0]	1, 4, 4	25.86550921	1.072222222	0.071481481	0.082126766	0.827863316	0.126193923	4
3	4	1	2	[2 4 0 4 3 1 0 1 0 1 0]	1, 4, 5	19.02260328	1.085555556	0.07237037	0.082069759	0.820289683	0.131136646	5
3	4	2	2	[2 4 0 4 3 1 0 1 0 1 0]	1, 4, 6	31.18219538	0.981111111	0.065407407	0.082077791	0.849417549	0.129240521	6
3	4	3	2	[2 4 0 4 3 1 0 1 0 1 0]	1, 4, 7	19.37709042	1.128888889	0.075259259	0.084755718	0.805928571	0.134419761	7
3	4	4	2	[2 4 0 4 3 1 0 1 0 1 0]	1, 4, 8	31.37125313	0.972222222	0.064814815	0.084908889	0.84856746	0.122726483	8
3	5	1	1	[4 2 0 2 3 2 0 1 0 1 0]	1, 5, 1	16.14878481	1.234444444	0.088174603	0.090637149	0.76158157	0.141604539	1
3	5	2	1	[4 2 0 2 3 2 0 1 0 1 0]	1, 5, 2	24.17861931	1.122222222	0.08015873	0.090104085	0.797537478	0.139618964	2
3	5	3	1	[4 2 0 2 3 2 0 1 0 1 0]	1, 5, 3	16.29136604	1.17	0.083571429	0.089422349	0.761068783	0.149142021	3
3	5	4	1	[4 2 0 2 3 2 0 1 0 1 0]	1, 5, 4	24.30962265	1.051111111	0.075079365	0.084416283	0.799803351	0.140048135	4
3	5	1	2	[4 2 0 2 3 2 0 1 0 1 0]	1, 5, 5	18.58298333	1.162222222	0.083015873	0.087475794	0.780379189	0.142791157	5
3	5	2	2	[4 2 0 2 3 2 0 1 0 1 0]	1, 5, 6	28.57492476	1.104444444	0.078888889	0.090969963	0.812799383	0.135752048	6
3	5	3	2	[4 2 0 2 3 2 0 1 0 1 0]	1, 5, 7	19.11268184	1.083333333	0.077380952	0.081442844	0.787478395	0.143971134	7
3	5	4	2	[4 2 0 2 3 2 0 1 0 1 0]	1, 5, 8	29.03934645	0.986666667	0.07047619	0.082518968	0.815895503	0.131801233	8
3	6	1	1	[4 4 0 2 3 2 0 1 0 1 0]	1, 6, 1	16.28617967	1.027777778	0.064236111	0.073174434	0.82022619	0.131653559	1
3	6	2	1	[4 4 0 2 3 2 0 1 0 1 0]	1, 6, 2	25.3509446	0.907777778	0.056736111	0.073737566	0.841133598	0.136284512	2
3	6	3	1	[4 4 0 2 3 2 0 1 0 1 0]	1, 6, 3	16.42755619	0.942222222	0.058888889	0.067512936	0.821298501	0.131982623	3
3	6	4	1	[4 4 0 2 3 2 0 1 0 1 0]	1, 6, 4	25.46784785	0.907777778	0.056736111	0.070422064	0.836387125	0.129046948	4
3	6	1	2	[4 4 0 2 3 2 0 1 0 1 0]	1, 6, 5	19.00517923	0.968888889	0.060555556	0.072722442	0.826541887	0.130972821	5
3	6	2	2	[4 4 0 2 3 2 0 1 0 1 0]	1, 6, 6	31.2096889	0.85	0.053125	0.070773021	0.854893739	0.128085214	6

Table 31: Output Data for Portfolio 3 Alternatives (continued).

ID.	FS.	Collab Matrix.	IOL RefTable.	FS	RP List Index	RPC	Num_ Blue Lost Mean	Blue % Lost Mean	Blue % Lost Std Dev	Red % Sup_ Mean	Red % Sup_ Std Dev	RP.
3	6	3	2	[4 4 0 2 3 2 0 1 0 1 0]	1, 6, 7	19.52983701	0.946666667	0.059166667	0.071769507	0.833391975	0.129437287	7
3	6	4	2	[4 4 0 2 3 2 0 1 0 1 0]	1, 6, 8	31.61160652	0.888888889	0.055555556	0.071752822	0.858939153	0.119599512	8
3	7	1	1	[4 2 0 2 3 1 0 1 0 1 0]	1, 7, 1	16.06773234	1.236666667	0.095128205	0.096832634	0.77625	0.145208064	1
3	7	2	1	[4 2 0 2 3 1 0 1 0 1 0]	1, 7, 2	23.93102003	1.184444444	0.091111111	0.09651919	0.802015432	0.137832612	2
3	7	3	1	[4 2 0 2 3 1 0 1 0 1 0]	1, 7, 3	16.13971527	1.306666667	0.100512821	0.100067442	0.767957672	0.143535995	3
3	7	4	1	[4 2 0 2 3 1 0 1 0 1 0]	1, 7, 4	23.99269581	1.208888889	0.092991453	0.09865545	0.806940476	0.136800582	4
3	7	1	2	[4 2 0 2 3 1 0 1 0 1 0]	1, 7, 5	18.32177708	1.223333333	0.094102564	0.095515895	0.787191799	0.145755837	5
3	7	2	2	[4 2 0 2 3 1 0 1 0 1 0]	1, 7, 6	27.82924754	1.111111111	0.085470085	0.096570549	0.820563492	0.138150727	6
3	7	3	2	[4 2 0 2 3 1 0 1 0 1 0]	1, 7, 7	18.59441758	1.223333333	0.094102564	0.098233604	0.787920194	0.143541681	7
3	7	4	2	[4 2 0 2 3 1 0 1 0 1 0]	1, 7, 8	28.05972645	1.08	0.083076923	0.096888695	0.826698854	0.131817672	8
3	8	1	1	[4 4 0 2 3 1 0 1 0 1 0]	1, 8, 1	16.2067484	1.054444444	0.070296296	0.080424416	0.829977954	0.128580264	1
3	8	2	1	[4 4 0 2 3 1 0 1 0 1 0]	1, 8, 2	25.1325417	0.906666667	0.060444444	0.077027973	0.848378307	0.129004797	2
3	8	3	1	[4 4 0 2 3 1 0 1 0 1 0]	1, 8, 3	16.27810149	1.11	0.074	0.082840142	0.816172399	0.13073356	3
3	8	4	1	[4 4 0 2 3 1 0 1 0 1 0]	1, 8, 4	25.18747562	1.041111111	0.069407407	0.081558145	0.837665344	0.128236963	4
3	8	1	2	[4 4 0 2 3 1 0 1 0 1 0]	1, 8, 5	18.76172652	1.036666667	0.069111111	0.08187006	0.833349206	0.132149834	5
3	8	2	2	[4 4 0 2 3 1 0 1 0 1 0]	1, 8, 6	30.60149438	0.975555556	0.065037037	0.08252181	0.850554674	0.127001223	6
3	8	3	2	[4 4 0 2 3 1 0 1 0 1 0]	1, 8, 7	19.03112445	1.057777778	0.070518519	0.082087327	0.83008157	0.127227265	7
3	8	4	2	[4 4 0 2 3 1 0 1 0 1 0]	1, 8, 8	30.79828463	0.917777778	0.061185185	0.076504901	0.85895194	0.127491364	8
3	9	1	1	[4 2 0 4 3 2 0 1 0 1 0]	1, 9, 1	18.14864832	0.896666667	0.056041667	0.071525408	0.854873016	0.123511859	1
3	9	2	1	[4 2 0 4 3 2 0 1 0 1 0]	1, 9, 2	29.06127792	0.843333333	0.052708333	0.070285092	0.873690035	0.119057004	2
3	9	3	1	[4 2 0 4 3 2 0 1 0 1 0]	1, 9, 3	18.30385864	0.902222222	0.056388889	0.0715863	0.851784392	0.12565431	3
3	9	4	1	[4 2 0 4 3 2 0 1 0 1 0]	1, 9, 4	29.17145817	0.815555556	0.050972222	0.068926492	0.866848325	0.118180368	4
3	9	1	2	[4 2 0 4 3 2 0 1 0 1 0]	1, 9, 5	20.93010354	0.823333333	0.051458333	0.067895258	0.864438713	0.120580189	5
3	9	2	2	[4 2 0 4 3 2 0 1 0 1 0]	1, 9, 6	33.74551965	0.74	0.04625	0.06542648	0.885056437	0.114850789	6

Table 32: Output Data for Portfolio 3 Alternatives (continued).

ID.	FS.	Collab Matrix.	IOL RefTable.	FS	RP List Index	RPC	Num_ Blue Lost Mean	Blue % Lost Mean	Blue % Lost Std Dev	Red % Sup_ Mean	Red % Sup_ Std Dev	RP.
3	9	3	2	[4 2 0 4 3 2 0 1 0 1 0]	1, 9, 7	21.51093999	0.92	0.0575	0.071672573	0.854547619	0.126049815	7
3	9	4	2	[4 2 0 4 3 2 0 1 0 1 0]	1, 9, 8	34.15815831	0.763333333	0.047708333	0.067692725	0.886946208	0.114186875	8
3	10	1	1	[2 2 0 2 3 2 0 1 0 1 0]	1, 10, 1	14.0109828	1.357777778	0.113148148	0.099389262	0.612768078	0.151212165	1
3	10	2	1	[2 2 0 2 3 2 0 1 0 1 0]	1, 10, 2	19.81709088	1.281111111	0.106759259	0.102491777	0.671650794	0.149249331	2
3	10	3	1	[2 2 0 2 3 2 0 1 0 1 0]	1, 10, 3	14.19609041	1.325555556	0.110462963	0.099731294	0.617622575	0.151778042	3
3	10	4	1	[2 2 0 2 3 2 0 1 0 1 0]	1, 10, 4	19.96724369	1.395555556	0.116296296	0.106965359	0.671121693	0.145454342	4
3	10	1	2	[2 2 0 2 3 2 0 1 0 1 0]	1, 10, 5	16.44106634	1.416666667	0.118055556	0.112990646	0.64833157	0.146340658	5
3	10	2	2	[2 2 0 2 3 2 0 1 0 1 0]	1, 10, 6	23.7546573	1.321111111	0.110092593	0.101061863	0.714440035	0.145892089	6
3	10	3	2	[2 2 0 2 3 2 0 1 0 1 0]	1, 10, 7	17.08511103	1.364444444	0.113703704	0.106939358	0.645387125	0.153464095	7
3	10	4	2	[2 2 0 2 3 2 0 1 0 1 0]	1, 10, 8	24.26159489	1.377777778	0.114814815	0.110180256	0.705703704	0.142036389	8
3	11	1	1	[2 4 0 2 3 1 0 1 0 1 0]	1, 11, 1	14.0705851	1.466666667	0.112820513	0.103578735	0.697580688	0.147391234	1
3	11	2	1	[2 4 0 2 3 1 0 1 0 1 0]	1, 11, 2	20.91885524	1.41	0.108461538	0.104958329	0.730774691	0.147521423	2
3	11	3	1	[2 4 0 2 3 1 0 1 0 1 0]	1, 11, 3	14.16347113	1.407777778	0.108290598	0.102214508	0.697358466	0.15766822	3
3	11	4	1	[2 4 0 2 3 1 0 1 0 1 0]	1, 11, 4	20.97791807	1.332222222	0.102478632	0.097883075	0.736142857	0.149767666	4
3	11	1	2	[2 4 0 2 3 1 0 1 0 1 0]	1, 11, 5	16.62373889	1.484444444	0.114188034	0.110609151	0.710866402	0.148310987	5
3	11	2	2	[2 4 0 2 3 1 0 1 0 1 0]	1, 11, 6	25.8705924	1.288888889	0.099145299	0.100511389	0.770450617	0.144708844	6
3	11	3	2	[2 4 0 2 3 1 0 1 0 1 0]	1, 11, 7	16.95440983	1.396666667	0.107435897	0.100527795	0.718667989	0.148660066	7
3	11	4	2	[2 4 0 2 3 1 0 1 0 1 0]	1, 11, 8	26.07581993	1.257777778	0.096752137	0.100029139	0.764075838	0.145528724	8
3	12	1	1	[2 2 0 4 3 1 0 1 0 1 0]	1, 12, 1	16.15464586	1.301111111	0.10008547	0.103305094	0.744508818	0.148673631	1
3	12	2	1	[2 2 0 4 3 1 0 1 0 1 0]	1, 12, 2	24.6360911	1.254444444	0.096495726	0.100898421	0.791919753	0.141922659	2
3	12	3	1	[2 2 0 4 3 1 0 1 0 1 0]	1, 12, 3	16.25199315	1.385555556	0.106581197	0.096171804	0.745193122	0.141238974	3
3	12	4	1	[2 2 0 4 3 1 0 1 0 1 0]	1, 12, 4	24.69393665	1.278888889	0.098376068	0.102074464	0.794564374	0.135334937	4
3	12	1	2	[2 2 0 4 3 1 0 1 0 1 0]	1, 12, 5	18.60290272	1.415555556	0.108888889	0.101089048	0.749720459	0.146855451	5

Table 33: Output Data for Portfolio 3 Alternatives (continued).

ID.	FS.	Collab Matrix.	IOL RefTable.	FS	RP List Index	RPC	Num_ Blue Lost Mean	Blue % Lost Mean	Blue % Lost Std Dev	Red % Sup_ Mean	Red % Sup_ Std Dev	RP.
3	12	2	2	[2 2 0 4 3 1 0 1 0 1 0]	1, 12, 6	28.46918122	1.18	0.090769231	0.096705102	0.806438272	0.13778324	6
3	12	3	2	[2 2 0 4 3 1 0 1 0 1 0]	1, 12, 7	18.96164872	1.287777778	0.099059829	0.100628429	0.765409171	0.140984035	7
3	12	4	2	[2 2 0 4 3 1 0 1 0 1 0]	1, 12, 8	28.68856241	1.13	0.086923077	0.099487931	0.820399471	0.133760795	8
3	13	1	1	[4 2 0 4 3 1 0 1 0 1 0]	1, 13, 1	18.01752145	0.92	0.061333333	0.080911871	0.860932099	0.127585509	1
3	13	2	1	[4 2 0 4 3 1 0 1 0 1 0]	1, 13, 2	28.79271391	0.825555556	0.055037037	0.075780852	0.875367284	0.121244826	2
3	13	3	1	[4 2 0 4 3 1 0 1 0 1 0]	1, 13, 3	18.09625205	0.951111111	0.063407407	0.082413504	0.860780864	0.122108334	3
3	13	4	1	[4 2 0 4 3 1 0 1 0 1 0]	1, 13, 4	28.84521427	0.886666667	0.059111111	0.078436079	0.868070106	0.118299156	4
3	13	1	2	[4 2 0 4 3 1 0 1 0 1 0]	1, 13, 5	20.49858322	0.885555556	0.059037037	0.074451389	0.860880071	0.119766426	5
3	13	2	2	[4 2 0 4 3 1 0 1 0 1 0]	1, 13, 6	32.91153397	0.844444444	0.056296296	0.076385476	0.882169753	0.115650183	6
3	13	3	2	[4 2 0 4 3 1 0 1 0 1 0]	1, 13, 7	20.80169728	0.886666667	0.059111111	0.079188818	0.862880071	0.127040625	7
3	13	4	2	[4 2 0 4 3 1 0 1 0 1 0]	1, 13, 8	33.11786474	0.792222222	0.052814815	0.073075204	0.881084656	0.117918053	8
3	14	1	1	[4 4 0 4 3 2 0 1 0 1 0]	1, 14, 1	18.26636695	0.571111111	0.031728395	0.051763053	0.904736772	0.102129724	1
3	14	2	1	[4 4 0 4 3 2 0 1 0 1 0]	1, 14, 2	30.1635077	0.588888889	0.032716049	0.051545049	0.90302425	0.107184367	2
3	14	3	1	[4 4 0 4 3 2 0 1 0 1 0]	1, 14, 3	18.42058527	0.601111111	0.033395062	0.054267572	0.901329806	0.104242968	3
3	14	4	1	[4 4 0 4 3 2 0 1 0 1 0]	1, 14, 4	30.26461218	0.594444444	0.033024691	0.053604737	0.910601852	0.099100219	4
3	14	1	2	[4 4 0 4 3 2 0 1 0 1 0]	1, 14, 5	21.2808094	0.622222222	0.034567901	0.057096283	0.906010582	0.108330204	5
3	14	2	2	[4 4 0 4 3 2 0 1 0 1 0]	1, 14, 6	36.40129943	0.627777778	0.034876543	0.055472239	0.908171517	0.106382018	6

Table 34: Output Data for Portfolio 3 Alternatives (continued).

ID.	FS.	Collab Matrix.	IOL RefTable.	FS	RP List Index	RPC	Num_ Blue Lost Mean	Blue % Lost Mean	Blue % Lost Std Dev	Red % Sup_ Mean	Red % Sup_ Std Dev	RP.
3	14	3	2	[4 4 0 4 3 2 0 1 0 1 0]	1, 14, 7	21.85888139	0.601111111	0.033395062	0.051874028	0.90200485	0.1054145	7
3	14	4	2	[4 4 0 4 3 2 0 1 0 1 0]	1, 14, 8	36.76642086	0.587777778	0.032654321	0.055896419	0.912820547	0.101137465	8
3	15	1	1	[4 4 0 4 3 1 0 1 0 1 0]	1, 15, 1	18.13744824	0.656666667	0.038627451	0.06011505	0.899343474	0.10891062	1
3	15	2	1	[4 4 0 4 3 1 0 1 0 1 0]	1, 15, 2	29.91894594	0.677777778	0.039869281	0.060423047	0.907105379	0.103842295	2
3	15	3	1	[4 4 0 4 3 1 0 1 0 1 0]	1, 15, 3	18.21564535	0.62	0.036470588	0.057064587	0.905197531	0.10503789	3
3	15	4	1	[4 4 0 4 3 1 0 1 0 1 0]	1, 15, 4	29.96705862	0.63	0.037058824	0.057190914	0.912119489	0.100803618	4
3	15	1	2	[4 4 0 4 3 1 0 1 0 1 0]	1, 15, 5	20.87225864	0.631111111	0.037124183	0.057984227	0.90572575	0.10211453	5
3	15	2	2	[4 4 0 4 3 1 0 1 0 1 0]	1, 15, 6	35.69632658	0.634444444	0.037320261	0.055792215	0.906003968	0.108459671	6
3	15	3	2	[4 4 0 4 3 1 0 1 0 1 0]	1, 15, 7	21.17305508	0.687777778	0.040457516	0.062203043	0.901938713	0.103937001	7
3	15	4	2	[4 4 0 4 3 1 0 1 0 1 0]	1, 15, 8	35.87688864	0.63	0.037058824	0.058455632	0.908946208	0.105054755	8
3	16	1	1	[2 4 0 4 3 2 0 1 0 1 0]	1, 16, 1	16.4549683	1.12	0.07	0.077070962	0.798095238	0.135475201	1
3	16	2	1	[2 4 0 4 3 2 0 1 0 1 0]	1, 16, 2	26.02253981	1.035555556	0.064722222	0.07762772	0.820905644	0.134577333	2
3	16	3	1	[2 4 0 4 3 2 0 1 0 1 0]	1, 16, 3	16.64336607	1.125555556	0.070347222	0.076441816	0.796955026	0.14136162	3
3	16	4	1	[2 4 0 4 3 2 0 1 0 1 0]	1, 16, 4	26.1328318	1.02	0.06375	0.077649484	0.824804674	0.135690861	4
3	16	1	2	[2 4 0 4 3 2 0 1 0 1 0]	1, 16, 5	19.52439869	1.117777778	0.069861111	0.077253286	0.806925926	0.134951454	5
3	16	2	2	[2 4 0 4 3 2 0 1 0 1 0]	1, 16, 6	31.77445729	0.964444444	0.060277778	0.071204913	0.844313492	0.124326599	6
3	16	3	2	[2 4 0 4 3 2 0 1 0 1 0]	1, 16, 7	20.19222493	1.065555556	0.066597222	0.078359485	0.808225309	0.134682067	7
3	16	4	2	[2 4 0 4 3 2 0 1 0 1 0]	1, 16, 8	32.16110159	0.908888889	0.056805556	0.0713167	0.835132716	0.131057612	8

Table 35: Alternative 1 Resource Processing Matrix.

	SOF Team 1	SOF Team 2	SOF Team 3	SOF Team 4	SOF Team 5	SOF Team 6	SOF Team 7	SOF Team 8	Intel Satellite	Central C2
SOF Team 1	0	2	2	2	2	2	2	2	2	2
SOF Team 2	2	0	2	2	2	2	2	2	2	2
SOF Team 3	2	2	0	2	2	2	2	2	2	2
SOF Team 4	2	2	2	0	2	2	2	2	2	2
SOF Team 5	2	2	2	2	0	2	2	2	2	2
SOF Team 6	2	2	2	2	2	0	2	2	2	2
SOF Team 7	2	2	2	2	2	2	0	2	2	2
SOF Team 8	2	2	2	2	2	2	2	0	2	2
Intel Satellite	2	2	2	2	2	2	2	2	0	3
Central C2	2	2	2	2	2	2	2	2	3	0

Table 36: Alternative 2 Resource Processing Matrix.

	SOF Team 1	SOF Team 2	SOF Team 3	SOF Team 4	SOF Team 5	Intel Satellite	Central C2
SOF Team 1	0	0	0	0	0	0	2
SOF Team 2	0	0	0	0	0	0	2
SOF Team 3	0	0	0	0	0	0	2
SOF Team 4	0	0	0	0	0	0	2
SOF Team 5	0	0	0	0	0	0	2
Intel Satellite	0	0	0	0	0	0	3
Central C2	2	2	2	2	2	3	0

Table 37: Alternative 3 Resource Processing Matrix.

	F/A-18 1	F/A-18 2	F/A-18 3	F/A-18 4	F/A-18 5	F/A-18 6	X-47B 1	X-47B 2	X-47B 3	X-47B 4	X-47B 5	X-47B 6	CVN
F/A-18 1	0	0	0	0	0	0	0	0	0	0	0	0	5
F/A-18 2	0	0	0	0	0	0	0	0	0	0	0	0	5
F/A-18 3	0	0	0	0	0	0	0	0	0	0	0	0	5
F/A-18 4	0	0	0	0	0	0	0	0	0	0	0	0	5
F/A-18 5	0	0	0	0	0	0	0	0	0	0	0	0	5
F/A-18 6	0	0	0	0	0	0	0	0	0	0	0	0	5
X-47B 1	0	0	0	0	0	0	0	0	0	0	0	0	5
X-47B 2	0	0	0	0	0	0	0	0	0	0	0	0	5
X-47B 3	0	0	0	0	0	0	0	0	0	0	0	0	5
X-47B 4	0	0	0	0	0	0	0	0	0	0	0	0	5
X-47B 5	0	0	0	0	0	0	0	0	0	0	0	0	5
X-47B 6	0	0	0	0	0	0	0	0	0	0	0	0	5
CVN	5	5	5	5	5	5	5	5	5	5	5	5	0

Table 38: Alternative 4 Resource Processing Matrix.

	F/A-18 1	F/A-18 2	X-47B 1	X-47B 2	CVN
F/A-18 1	0	2	3	3	5
F/A-18 2	2	0	3	3	5
X-47B 1	3	3	0	2	5
X-47B 2	3	3	2	0	5
CVN	5	5	5	5	0

Table 39: Alternative 5 Resource Processing Matrix.

	F/A-18 1	F/A-18 2	F/A-18 3	F/A-18 4	AH-64 1	AH-64 2	AH-64 3	AH-64 4	EA-6B 1	EA-6B 2	EA-6B 3	EA-6B 4	M252 1	M252 2	M252 3	DDG	E-2	CVN
F/A-18 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5
F/A-18 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5
F/A-18 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5
F/A-18 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5
AH-64 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
AH-64 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
AH-64 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
AH-64 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
EA-6B 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	5
EA-6B 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	5
EA-6B 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	5
EA-6B 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	5
M252 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1
M252 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1
M252 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1
DDG	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1
E-2	2	2	2	2	1	1	1	1	2	2	2	2	2	2	2	0	0	5
CVN	5	5	5	5	1	1	1	1	5	5	5	5	1	1	1	1	5	0

Table 40: Alternative 6 Resource Processing Matrix.

	F/A-18 1	F/A-18 2	AH-64 1	AH-64 2	EA-6B 1	EA-6B 2	EA-6B 3	EA-6B 4	M252 1	M252 2	M252 3	DDG	E-2	CVN
F/A-18 1	0	2	2	2	2	2	2	2	1	1	1	2	3	5
F/A-18 2	2	0	2	2	2	2	2	2	1	1	1	2	3	5
AH-64 1	2	2	0	2	2	2	2	2	1	1	1	0	3	1
AH-64 2	2	2	2	0	2	2	2	2	1	1	1	0	3	1
EA-6B 1	2	2	2	2	0	2	2	2	2	2	2	2	3	5
EA-6B 2	2	2	2	2	2	0	2	2	2	2	2	2	3	5
EA-6B 3	2	2	2	2	2	2	0	2	2	2	2	2	3	5
EA-6B 4	2	2	2	2	2	2	2	0	2	2	2	2	3	5
M252 1	1	1	1	1	2	2	2	2	0	2	2	0	2	2
M252 2	1	1	1	1	2	2	2	2	2	0	2	0	2	2
M252 3	1	1	1	1	2	2	2	2	2	2	0	0	2	2
DDG	2	2	0	0	2	2	2	2	0	0	0	0	2	2
E-2	3	3	3	3	3	3	3	3	2	2	2	2	0	5
CVN	5	5	1	1	5	5	5	5	2	2	2	2	5	0

REFERENCES

[1] "Defense acquisition guidebook." Internet Publication. Last Accessed October 20, 2010. [Online]. Available: https://dag.dau.mil/Pages/Default.aspx.

[2] "Federation of american scientists webpage." [Online]. Last Accessed October 3, 2011. Available: www.fas.org.

[3] "Merriam-webster online dictionary." Last Accessed August 8, 2011. [Online]. Available: www.merriam-webster.com/dictionary.

[4] "Oxford online dictionary." Last Accessed August 8, 2011. [Online]. Available: http://oxforddictionaries.com.

[5] "Top five systems engineering issues in defense (version 8)," August 18 2006. Last Accessed October 22, 2010. [Online]. Available: http://www.ndia.org/Resources/Pages/Publication_Catalog.aspx.

[6] "Modular space: Darpa's f6 program," April 27 2010.

[7] *Vision and Voyages for Planetary Science in the Decade 2013-2022*. 500 Fifth Street, N.W., Washington, DC 20001: The National Academies Press, National Research Council, 2011. Prepublication Copy Subject to Further Editorial Correction.

[8] Ackerman, F., "Critique of cost-benefit analysis, and alternative approaches to decision-making: A report to friends of the earth, england, wales and northern island," January 2008. Last Accessed October 24, 2010. [Online]. Available: http://www.ase.tufts.edu/gdae/Pubs/rp/Ack_UK_CBAcritique.pdf.

[9] Acquisition, Logistics & Technology Enterprise Systems & Services, *Probability of Program Success Operations Guide*, August 22 2007.

[10] Air Force Studies Board, Division on Engineering and Physical Sciences, Committe on Pre-Milestone A Systems Engineering, *Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Acquisition*, 2008.

[11] Alberts, D. S., Garstka, J. J., Hayes, R. E., and Signori, D. A., *Understanding Information Age Warfare*. CCRP, August 2001.

[12] Alberts, D. S., Garstka, J. J., and Stein, F. P., *Network Centric Warfare: Developing and Leveraging Information Superiority*. CCRP Publication Series, DoD C4ISR Cooperative Research Program, 2nd (Revised) ed., Aug. 1999.

[13] Alderson, D. and Doyle, J., "Contrasting views of complexity and their implications for network-centric infrastructures," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, pp. 839–852, July 2010.

[14] Andrews, R., "An overview of acquisition logistics." Source Material: Presentation to Defense Acquisition University, Fort Belvoir, VA., 2003. Image from Industry's Obligation for Mission Success (Liang, Alex). 19th AIAA Space Flight Mechanics Conference. Last Accessed October 22, 2010. [Online]. Available: `www.space-flight.org/AAS_meetings/2009_winter/LIANG_AIAA.ppt`.

[15] Arnold, S. A., Harmon, B. R., Tyson, K. W., Fasana, K. G., and Wait, C. S., "Defense department profit and contract finance policies and their effects on contract and contractor perfo." IDA Paper P-4284 (Revised), February 2009. [Online]. Last Accessed 27 September 2011. Available at: `http://www.acq.osd.mil/mibp/docs/ida_paper_p-4284_revised.pdf`.

[16] Assessment Panel of the Defense Acquisition Performance Assessment Project, *Defense Acquisition Performance Assessment Report*, January 2006.

[17] Bagdatli, B., Griendling, K., Kalpakchian, D., Jones, E., Ussery, S., Ball, J., Kallman, J., and Mavris, D., "A method for examining the impact of interoperability on mission performance in a system-of-systems," in *Aerospace Conference, 2010 IEEE*, pp. 1 –15, march 2010.

[18] Bak, P., Tang, C., and Wiesenfeld, K., "Self-organized criticality: An explanation of 1/f noise," *Physical Review Letters*, vol. 59, pp. 381–384, July 27 1987.

[19] Balakrishnan, R., "The energy of a graph," *Linear Algebra and its Applications*, vol. 387, pp. 287–295, 2004.

[20] Baldwin, K., Dahmann, J., and Gullifer, R., "Defense View on Considerations for System of Systems SE." Presentation, October 25 2006. Last Accessed October 19, 2010. [Online]. Available: `http://www.dtic.mil/ndia/2006systems/Wednesday/baldwin.pdf`.

[21] Balestrini-Robinson, S., *A Modeling Process to Understand Complex System Architectures*. PhD thesis, Georgia Institute of Technology, August 2009.

[22] Bar-Yam, Y., *Dynamics of Complex Systems*. Studies in Nonlinearity, Reading, Massachusetts: Addison-Wesley, 1997.

[23] Baratta, J., "Department of the navy probability of program success v2.0 brief." Presentation, February 19 2010. [Online]. Last Accessed 3 October 2011. Available: `https://www.ncca.navy.mil/dodcas/briefings/DODCAS2010-POPS_v2.pdf`.

[24] Bard, M., "Probability of program success portfolio analysis." U.S. Air Force Presentation, May 20 2009. Last Accessed November 11, 2009. [Online].

[25] Barkdoll, T. C., Gaver, D. P., Glazebrook, K. D., Jacobs, P. A., and Posadas, S., "Suppression of enemy air defenses (sead) as an information duel," *Naval Research Logistics*, vol. 49, pp. 723–742, June 2002.

[26] Bearden, D. A., "A complexity-based risk assessment of low-cost planetary missions: When is a mission too fast and too cheap?," in *Fourth IAA International Conference on Low-Cost Planetary Missions*, (Laurel, MD), Johns Hopkins University/ Applied Physics Laboratory, May 2-5 2000.

[27] Bearden, D. A., "Small-satellite costs," *Crosslink*, pp. 33–44, Winter 2000/2001.

[28] Becz, S., Pinto, A., Zeidner, L. E., Khire, R., Banaszuk, A., and Reeve, H. M., "Design system for managing complexity in aerospace systems," in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2010.

[29] Bennett, C. H., *Logical Depth and Physical Complexity.* Oxford University Press, 1988.

[30] Berka, K., *Measurement: Its Concepts, Theories and Problems*, vol. 72 of *Boston Studies in the Philosophy of Science*. Boston, USA: D. Reidel Publishing Company, 1983.

[31] Biltgen, P. T., *A Methodology for Capability-Based Technology Evaluation for System-of-Systems.* PhD thesis, Georgia Institute of Technology, May 2007.

[32] Binder, P., "A phase diagram for elementary cellular automata," *Complex Systems*, vol. 7, pp. 241–247, 1993.

[33] Boardman, J. and Sauser, B., "System of systems - the meaning of *of*," in *IEEE/SMC International Confefence on System of Systems Engineering*, (Los Alamitos, CA, USA), pp. 118–123, IEEE Computer Society, April 2006.

[34] Boardman, J., DiMario, M., Sauser, B., and Verma, D., "System of systems characteristics and interoperability in joint command and control." Presentation at 2nd Annual System of Systems Engineering Conference, Defense Acquisition University, Ft. Belvoir, VA, July 25-26 2006. Last Accessed October 18, 2010. [Online]. Available: `http://www.sosece.org/pdfs/2ndConference/Presentations/SoSECE%20Howell%20Day%20One/DiMario.pdf`.

[35] Bonabeau, E., "Agent-based modeling: Methods and techniques for simulating human systems," in *Proceedings of the National Academies of Science*, vol. 90, pp. 7280–7287, May 14 2002.

[36] Braha, D., Minai, A., and Bar-Yam, Y., eds., *Complex Engineered Systems: Science Meets Technology*. Understanding Complex Systems, Springer: Complexity, Springer, 2006.

[37] Breyfogle III, F. W., *Implementing Six Sigma: Smarter Solutions Using Statistical Methods*. John Wiley & Sons, Inc., 2 ed., 2003.

[38] Brown, B., *Introduction to Defense Acquisition Management*. Fort Belvoir, Virginia: Defense Acquisition University Press, tenth ed., August 2010.

[39] Brown, O. C., Eremenko, P., and Collopy, P. D., "Value-centric design methodologies for fractionated spacecraft: Program summary from phase 1 of the darpa system f6 program," No. AiAA 2009-6540, September 2009. Pasadena, California.

[40] C4ISR Architectures Working Group, *Levels of Information Systems Interoperability (LISI)*. Department of Defense, March 30 1998.

[41] Canada, J. R., Sullivan, W. G., and White, J. A., *Capital Investment Analysis for Engineering and Management*. Upper Saddle River, New Jersey: Prentice Hall, Inc., second ed., 1996.

[42] Cares, J., *Distributed Network Operations: The Foundation of Network Centric Warfare*. Newport, RI: Alidade Press, 2005.

[43] Carlstrom, G., "2010 dod budget request includes cuts," *Defense News*, May 7 2009. Last Accessed October 21, 2010. [Online]. Available: `http://www.defensenews.com/story.php?i=4078101`.

[44] Cebrowski, A. K., "Network centric warfare: Its origin and future," in *Proceedings of the Naval Institute*, vol. 124, Number 1, pp. 28–35, January 1998.

[45] Chairman of the Joint Chiefs of Staff (CJCS), *CJCSI 3170.01F: Operation of the Joint Capabilities Integration and Development System*, May 1 2007.

[46] Chairman of the Joint Chiefs of Staff (CJCS), *CJCSI 3170.01G: Operation of the Joint Capabilities Integration and Development System*, March 1 2009.

[47] Chief of Naval Operations, Commandant of the Marine Corps, and Commandant of the United States Coast Guard, *Universal Naval Task List (UNTL), Version 3.0, OPNAVINST 3500.38B/ MCO3500.26A/ USCG COMDTINST 3500.1B*, January 30 2007.

[48] Churchman, C. W. and Ratoosh, P., eds., *Measurement: Definitions and Theories*. New York: John Wiley & Sons, Inc., 1959.

[49] Colby, G., "Scalability lessons learned: The key to open architecture success." SPIES Briefing, March 25 2009. Last Accessed October 19, 2010. [Online]. Available: `www.sae.org/servlets/works/committeeResources.do?resourceID=134011`.

[50] Conklin, J., *Wicked Problems and Social Complexity.* CogNexus Institute, 2001.

[51] Cox, J. C. and Rubinstein, M., *Options Markets.* Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1985.

[52] Crawley, E., "Massachusetts institute of technology system architecture course notes," 2005.

[53] Dagli, C. and Kilicay-Ergin, N., *System of Systems Engineering: Principles for the 21st Century*, ch. 4. System of systems architecting, pp. 77–100. Wiley, 2009.

[54] Dahmann, J., Rebovich, G., Lowry, R., Lane, J., and Baldwin, K., "An implementers' view of systems engineering for systems of systems," in *Systems Conference (SysCon), 2011 IEEE International*, pp. 212 –217, april 2011.

[55] Das, B., "Effects-based operations: Simulations with cellular automata," Tech. Rep. DSTO-RR-0275, Defence Science and Technology Organisation (Australia), 2004.

[56] Davis, P. K. and Dreyer, P., "Rand's portfolio analysis tool (pat): Theory, methods, and reference manual." RAND Corporation. Document Number TR-756-OSD, 2009.

[57] Davis, P. K., Shaver, R. D., and Beck, J., "Portfolio-analysis methods for assessing capability options." RAND Corporation. Document Number MG-662-OSD, 2008.

[58] de Neufville, R., "Architecting/designing engineering systems using real options." Massachusetts Institute of Technology Engineering Systems Division Working Series Paper ESD-WP-2003.01.09-ESD Internal Symposium, May 29-30 2002.

[59] DeLaurentis, D. and Callaway, R., "A system-of-systems perspective for future public policy decisions," *Review of Policy Research*, vol. 21, Issue 6, pp. 829–837, November 2004.

[60] Department of Defense, *DoD Directive 5000.01, "The Defense Acquisition System"*, May 12 2003.

[61] Department of Defense, *Risk Management Guide for DoD Acquisition*, sixth ed., August 2006.

[62] Department of Defense, *DoD Architecture Framework Version 1.5*, April 23 2007.

[63] Department of Defense, *DoD Directive 5000.02, "Operation of The Defense Acquisition System"*, December 8 2008.

[64] Department of Defense, *DoD Architecture Framework Version 2.0*, May 18 2009.

[65] Department of Defense Systems Management College, Defense Acquisition University Press, Fort Belvoir, Virginia, *Systems Engineering Fundamentals*, January 2001.

[66] Department of the Navy, Assistant Secretary of the Navy Research Development and Acquisition, *Naval PoPS: Program Health Assessment Guidance for Naval Acquisition Programs*, September 2008.

[67] Dickerson, C. and Mavris, D., *Architecture and Principles of Systems Engineering.* CRC Complex and Enterprise Systems Engineering, Auerbach Publications, April 2009.

[68] Dickerson, C., Soules, S., Sabins, M., and Charles, P., *Using Architectures for Research, Development, and Acquisition.* Office of the Chief Engineer of the Navy, Assistant Secretary of the Navy. Defense Technical Information Center (DTIC): (www.dtic.mil), 2003. AD Number ADA427961.

[69] Director Systems and Software Engineering, Deputy Under Secretary of Defense (Acquisition and Technology), Office of the Under Secretary of Defense (Acquisition, Technology and Logistics), *Systems Engineering Guide for Systems of Systems*, August 2008.

[70] Dixon, D. D., "Non-deterministic chaos." Last Accessed October 25, 2010. [Online]. Available: `http://arxiv.org/abs/chao-dyn/9408001`, August 8 1994.

[71] Domercant, J. C. and Mavris, D. N., "Measuring the architectural complexity of military systems-of-systems," in *2011 IEEE Aerospace Conference*, IEEE Aerospace Conference (Big Sky, MT), pp. 1–16, March 5-12 2011. Paper No. 1649.

[72] Dooley, K., *Organizational Complexity*, pp. 5013–5022. London: Thompson Learning, 2002.

[73] Dori, D., *Object-process methodology: a holistics systems paradigm.* Springer Verlag, 2002.

[74] E.C. Aldridge, J., "Evolutionary acquisition and spiral development." Memorandum for Secretaries of the Military Departments, CJCS, Under Secretaries of Defense, Assistant Secretaries of Defense, Inspector General, DoD General Counsel, DoD Directors of the Defense Agencies, April 12 2002.

[75] Edmonds, B., *Syntactic Measures of Complexity.* Doctor of philosophy in the faculty of arts, University of Manchester, 1999.

[76] Edmonds, B., "What is complexity? the philosophy of complexity per se with applications to some examples in evolution," *The Evolution of Complexity*, p. 296, 1999.

[77] Eremenko, P., "META: Novel methods for design and verification of complex systems." Industry Day Briefing, December 22 2009. Last Accessed 23 September 2011.

[78] Fiedler, M., "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, pp. 298–305, April 14 1973.

[79] Frits, A. P. and Mavris, D. N., "Use of probability of success as an independent variable for decision making," in *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, AIAA 2004-6206, September 20-22 2004.

[80] Fulghum, D. A. and Wall, R., "Young out, carter in: Sasc once more demands additional defense acquisition changes," *Aviation Week & Space Technology*, vol. 170, pp. 24–27, March 2 2009.

[81] Gates, J., "Cost as an independent variable." Teaching Note, April 2006. Last Accessed October 20, 2010. [Online]. Available: `https://acc.dau.mil/adl/en-US/30380/file/5380/06%20-%20CAIV%20TN%20-%20Apr%2006.pdf`.

[82] Gell-Mann, M. and Lloyd, S., "Information measures, effective complexity, and total information," *Complexity*, vol. 2, no. 1, pp. 44–52, 1996.

[83] German, B. J. and Daskilewicz, M. J., "An MDO-inspired systems engineering perspective for the "wicked" problem of aircraft conceptual design," in *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, AIAA 2009-7115, AIAA, 21-23 September 2009. Hilton Head, South Carolina.

[84] Gershenson, C., "Introduction to random boolean networks," in *Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems* (Bedau, M., Husbands, P., Hutton, T., Kumar, S., and Suzuki, H., eds.), vol. ALife IX, pp. 160–173, 2004.

[85] Government Accountability Office, *Cost Assessment Guide: Best Practices for Estimating and Managing Program Costs (Exposure Draft)*, July 2007. Report Number GAO-07-1134SP.

[86] Government Accountability Office, *Defense Acquisitions: Assessments of Selected Weapon Programs*, March 2009. Report Number GAO-09-326SP.

[87] Government Accountability Office, *Defense Acquisitions: Assessments of Selected Weapon Programs*, March 2009. Report Number GAO-09-326SP.

[88] Grinstead, C. M. and Shell, L. J., *Introduction to Probability.* American Mathematical Society, 2006.

[89] Gusfield, D., "Computing the strength of a graph," *Society for Industrial and Applied Mathematics*, vol. 20, pp. 639–654, August 1991.

[90] Hartcup, G., *The War of Inventions: Scientific Developments, 1914-1918.* Potomac Books, Inc., 1988.

[91] Hayter, A. J., *Probability and Statistics for Engineers and Scientists.* Duxbury (Thomson, Brooks/Cole), third ed., 2007.

[92] Hazelrigg, G. A., *Systems Engineering: An Approach to Information-Based Design.* Prentice Hall International Series In Industrial and Systems Engineering, Prentice Hall, Inc., 1996.

[93] J-8, "Introduction to the joint capabilities integration & development system." Presentation (Adapted from presentation by Major General Robin Scott).

[94] Joint Chiefs of Staff, *Joint Publication 1-02: Department of Defense Dictionary of Military and Associated Terms*, April 12 2001. (As Amended Through 31 July 2010).

[95] Joint Chiefs of Staff, *Joint Publication 2-03: Geospatial Intelligence Support to Joint Operations.* Department of Defense, March 22 2007.

[96] Joint Chiefs of Staff, *Joint Publication 3-01: Countering Air and Missile Threats*, 05 February 2007.

[97] Joint Chiefs of Staff, *Joint Publication 1: Doctrine for the Armed Forces of the United States*, March 20 2009.

[98] Jones, R. R., Hardin, P., and Irvine, A., "Simpleparametric model for estimating development (rdt&e) cost for large-scale systems." Presentation at 2009 ISPA/SCEA Joint Conference, 2009. Last Accessed September 23, 2011.

[99] Joslyn, C. and Rocha, L., "Towards semiotic agent-based models of socio-technical organizations," in *AI, Simulation and Planning in High Autonomy Systems (AIS 2000)*, pp. 70–79, 2000.

[100] Kinnunen, M. J., "Complexity measures for system architectures," Master's thesis, Massachussetts Institute of Technology, February 2006.

[101] Kinsner, W., "Complexity & its measures in cognitive and other complex systems," in *7th IEEE International Conference on Cognitive Informatics* (Wang, Y., Zhang, D., Latombe, J., and Kinsner, W., eds.), pp. 13–29, 2008.

[102] Kinzel, W., "Phase transitions of cellular automata," *Zeitschrift für Physik B Condensed Matter*, vol. 58, no. 3, pp. 229–244, 1985.

[103] Kirkwood, C. W., *System Dynamics Methods: A Quick Introduction*. Arizona State University College of Business, 2008.

[104] Langton, C. G., "Computation at the edge of chaos: Phase transitions and emergent computation," *Physica D*, vol. 42, pp. 12–37, 1990.

[105] Larson, W. J. and Wertz, J. R., eds., *Space Mission Analysis And Design*. Space Technology Series, Microcosm, Inc. & Kluwer Academic Publishers, second ed., 1992.

[106] Levin, H. M. and McEwan, P. J., *Cost-Effectiveness Analysis: Methods and Applications*. Sage Publications, Inc., 2 ed., 2001.

[107] Li, W., Packard, N. H., and Langton, C., "Transition phenomena in cellular automata rule space," *Physica D*, vol. 45, pp. 77–94, 1990.

[108] Ling, M. F., Moon, T., and Kruzins, E., "Proposed network centric warfare metrics: From connectivity to the ooda cycle," *MORS Journal*, vol. 10, pp. 1–20, April 2005.

[109] Lloyd, S., "Measures of complexity: A nonexhaustive list," *FeedForward: IEEE Control Systems Magazine*, pp. 7–8, August 2001.

[110] Lloyd, S. and Pagels, H., "Complexity as thermodynamic depth," *Annals of Physics*, vol. 188, no. 1, pp. 186 – 213, 1988.

[111] Luehrman, T. A., "Investment opportunities as real options: Getting started on the numbers," *Harvard Business Review*, pp. 4–15, July-August 1998. Reprint Number 98404.

[112] Luehrman, T. A., "Strategy as a portfolio of real options," *Harvard Business Review*, pp. 89–99, September-October 1998. Reprint Number 98506.

[113] Maier, M. and Rechtin, E., *The Art of Systems Architecting*. CRC Press, 2 ed., 2002.

[114] Maier, M., "Architecting principles for system of systems," *Systems Engineering*, vol. 1, No. 4, pp. 267–284, 1998.

[115] Martin, I. F., *Valuation of Design Adaptability in Aerospace Systems*. PhD thesis, Georgia Institute of Technology, April 2008.

[116] Mavris, D. N. and Bandte, O., "A probabilistic approach to multivariate constrained robust design simulation," *SAE Technical Paper*, no. 975508, 1997.

[117] Mavris, D. N. and DeLaurentis, D. A., "A probabilistic approach for examining aircraft concept feasibility and viability," *Aircraft Design*, vol. 3, pp. 79–101, 2000.

[118] McCabe, T. J., "A complexity measure," *IEEE Transactions on Software Engineering*, vol. SE-2, No. 4, pp. 308–320, December 1976.

[119] McCabe, T. J. and Butler, C. W., "Design complexity measurement and testing," *Communications of the ACM*, vol. 32, pp. 1415–1425, December 1989.

[120] McNaugher, T. L., "Weapons procurement: The futility of reform," *International Security*, vol. 12, no. 2, p. 67, Autumn 1987.

[121] Melnyk, R., "Petri nets: An alternative to markov chains," *The Journals of the Reliability Information Analysis Center*, vol. 1st Quarter, 2004, pp. 7–11, 2004.

[122] Meyer, C. D., *Matrix Analysis and Applied Linear Algebra.* Philadelphia, PA: Society for Industrial and Applied Mathematics, 2000.

[123] Mitchell, M., *Complexity: A Guided Tour.* Oxford University Press, 2009.

[124] Moffat, J., *Complexity Theory and Network Centric Warfare.* Information Age Transformation Series, CCRP, 2003.

[125] Murphy, R. and Birkler, J., "Dollar value and risk levels: Changing how weapon system programs are managed," occasional paper, The RAND Corporation, 2009.

[126] Newman, M. E. J., "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, June 2003.

[127] Office of Aerospace Studies, Air Force Materiel Command (OAS/A9), *Office of Aerospace Studies: Analysis of Alternatives (AoA) Handbook*, July 2008.

[128] Office of Management and Budget, *Terminations, Reductions, and Savings: Budget of the U.S. Government Fiscal Year 2011*, May 7 2010.

[129] Office of the Under Secretary of Defense (Acquisition and Technology), *DoD Guide to Integrated Product and Process Development (Version 1.0)*, February 5 1996.

[130] Okon, W., "DoD-CIO/ASD-NII enterprise architecture & standards directorate presentation to DAU IRM Class Level III." Presentation, July 28 2008. Last Accessed October 19, 2010. [Online]. Available: `https://acc.dau.mil/adl/en-US/225556/file/36509/OSDDoDAFBriefing.ppt`.

[131] Olagbemiro, A., Mun, J., and Shing, M.-T., "Application of real options theory to dod software acquisitions." Naval Postgraduate School Acquisition Research Sponsored Report Series: NPS-AM-09-007, February 20 2009.

[132] Pahl, G., Beitz, W., Feldhusen, J., and Grote, K. H., *Engineering Design: A Systematic Approach.* London: Springer-Verlag, 3rd ed., 2007.

[133] Panconesi, A., *The Stationary Distribution of a Markov Chain*. Rome: La Sapienza University of Rome, 2005.

[134] Pennock, M. J., Rouse, W. B., and Kollar, D. L., "Transforming the acquisition enterprise: A framework for analysis and a case study of ship acquisition," *Systems Engineering*, vol. 10, no. 2, pp. 99–117, 2007.

[135] Peoples, R. and Wilcox, K., "Value-based multidisciplinary optimization for commercial aircraft design and business risk assessment," *Journal of Aircraft*, vol. 43, pp. 913–921, July-August 2006.

[136] Perrow, C., *Normal Accidents*. New York: Basic Books, Inc., 1984.

[137] Perry, W., Button, R. W., Bracken, J., and Thomas Sullivan, J. M., *Measures of Effectiveness for the Information-Age Navy: The Effects of Network-Centric Operations on Combat Outcomes*. RAND, 2002.

[138] Perry, W., Signori, D., and Boon, J., "Exploring information superiority," tech. rep., RAND, 2004.

[139] Raczynski, C. M., Kirby, M. R., and Mavris, D., "A dynamic process for strategic roadmapping and technology portfolio management," No. AIAA 2006-7789, (Wichita, Kansas), 25-27 September 2006.

[140] Redman, Q. and Stratton, G., "Why affordability is a systems engineering metric," in *Digital Avionics Systems, 2001. DASC. The 20th Conference*, vol. 1, pp. 4B3.1–4B3.12, 2002.

[141] Rich, M. and Dews, E., "Improving the military acquisition process." A Project AIR FORCE report prepared for the United States Air Force. RAND Corporation, February 1986. Report Number R-3373-AF/RC.

[142] Robinson, J., *The Economics of Imperfect Competition*. London: Macmillan, 1933.

[143] Rouse, W., "Complex engineered, organizational and natural systems," *Systems Engineering*, vol. 10, no. 3, pp. 260–271, 2007.

[144] Sauser, B., Ramirez-Marquez, J. E., Magnaye, R., and Tan, W., "A systems approach to expanding the technology readiness level within defense acquisition," *International Journal of Defense Acquisition Management*, vol. 1, pp. 39–58, 2008.

[145] Savage, G. T., Nix, T. W., and Whitehead, C. J., "Strategies for assessing and managing organizational stakeholders," *The Executive*, vol. 5, pp. 61–75, May 1991.

[146] Schwartz, M., "Defense acquisitions: How dod acquires weapon systems and recent efforts to reform the process." Congressional Research Service Report for Congress, July 10 2009.

[147] Shannon, C. E. and Weaver, W., *The mathematical theory of communication.* University of Illinois Press, Urbana, 1949.

[148] Smith, D., "Strategic roadmaps." Technology Futures, Inc. White Paper, 2005.

[149] Smuts, J., *Holism and Evolution.* The MacMillan Company, 1926.

[150] Solomon, P. J., "Integrating systems engineering with earned value management," *Defense Acquisition, Technology & Logistics Magazine*, vol. May-June 2004, pp. 42–46, 2004.

[151] Suh, N. P., *Complexity: Theory and Applications.* MIT- Pappalardo Series in Mechanical Engineering, Oxford University Press, 2005.

[152] Summers, J. D. and Shah, J. J., "Mechanical engineering design complexity metrics: Size, coupling, and solvability," *Journal of Mechanical Design*, vol. 132, pp. 021004–1 – 021004–11, February 2010.

[153] Sussman, J. M., "Collected views on complexity in systems." MIT Engineering Systems Division ESD-WP-2003-01.06, May 29-30 2002.

[154] Thorn, E., "Probability of program success." U.S. Air Force Presentation, June 6 2007. Last Accessed October 25, 2010. [Online]. Available: `www.acc.dau.mil`.

[155] Tsonis, A. A., *Chaos: From Theory to Applications.* New York: Plenum Press, 1992.

[156] Tyson, K. W., Harmon, B. R., and Utech, D. M., "Understanding cost and schedule growth in acquisition programs." Institute for Defense Analyses Paper P-2967, July 1994.

[157] Urwin, E. N., Venters, C. C., Russell, D. J., Liu, L., Luo, Z., Webster, D. E., Henshway, M., and Xu, J., "Scenario-based design and evaluation for capability," in *5th IEEE International Conference on System of Systems Engineering (SoSE2010)*, 2010.

[158] U.S. Marine Corps, *MCWP 3-22.2: Suppression of Enemy Air Defenses (SEAD)*, May 18 2001.

[159] Valerdi, R., Boehm, B. W., and Reifer, D. J., "Cosysmo: A constructive systems engineering cost model coming of age," in *INCOSE 2003-13th Annual International Symposium Proceedings*, June 29 - July 3 2003. Track: Modeling/Cost.

[160] Volovoi, V., "Modeling of system reliability using petri nets with aging tokens," *Reliability Engineering and System Safety*, vol. 84, pp. 149–161, 2003.

[161] WARWICK, G. and NORRIS, G., "Calls escalate for revamp of systems engineeering process," *Aviation Week & Space Technology*, p. 72, November 1, 2010.

[162] WEAVER, W., "Science and complexity," *American Scientist*, vol. 36, pp. 536–544, 1948.

[163] WOLFRAM, S., "Cellular automata as models of complexity," *Nature*, vol. 311, pp. 419–424, October 1984.

[164] WOLFRAM, S., *Theory and Applications of Cellular Automata (including selected papers 1983 - 1986)*, vol. 1 of *Advanced series on complex systems*. World, 1986.

[165] WOLLOVER, D., "Design-to-cost decision support system (DTC-DSS)," *Journal of Cost Analysis and Management*, pp. 3–33, December 1999.

[166] ZEIDNER, L. E., BECZ, S., REEVE, H. M., and KHIRE, R., "Architectural enumeration & evaluation for identification of low-complexity systems." Last Accessed October 17, 2010.

[167] ZEIDNER, L. E., BECZ, S. B., REEVE, H. M., and KHIRE, R., "Design issues for a bottom-up complexity metric applied to hierarchical systems." Last Accessed October 17, 2010.

[168] ZHAO, J., "On assessing the complexity of software architectures," in *Proceedings of the third international workshop on Software architecture*, (New York, NY, USA), pp. 163–166, ACM Press, 1998.

[169] ZIMMERMANN, H., "OSI reference model: The ISO model of architecture for open systems interconnection," *IEEE Transactions on Communications*, vol. 28, pp. 425–432, April 1980.

VITA

Jean Charles Domerçant was born December 25, 1976 in Brooklyn, NY to Jean-Claude and Marie Domerçant. He attended Bolingbrook High School in Bolingbrook, IL before enrolling at the University of Illinois at Urbana-Champaign in 2000. As an undergraduate, Mr. Domerçant conducted undergraduate research as a Ronald E. McNair scholar in the area of spacecraft trajectory optimization. After receiving a B.S. Aeronautical/Astronautical Engineering, Mr. Domerçant received a commission as an officer in the U.S. Navy's submarine force, serving from 2000-2006. During Mr. Domerçant's time in the U.S. Navy, he served aboard the USS Los Angeles (SSN 688) where he completed his qualifications in submarines and as a naval nuclear engineer. He also received a Master of Engineering Management from Old Dominion University in May 2006. Later that year Mr. Domerçant received an honorable discharge from naval service and enrolled in the graduate program at the Georgia Institute of Technology, receiving a Master of Science in Aerospace Engineering in 2008. Mr. Domerçant also participated in the professional internship program during his time as a graduate researcher at the Aerospace Systems Design Laboratory at Georgia Tech. Over the past five years, Mr. Domerçant has worked with a number of industry and government sponsors in the areas of missile defense, technology portfolio selection, defense acquisitions and decision making, and architecture-based systems-of-systems engineering.